The International Congress for global Science and Technology

ICGST International Journal on Graphics, Vision and Image Processing (GVIP)

Volume (17), Issue (II)
December 2017

www.icgst.com
www.icgst-amc.com
www.icgst-ees.com

GVIP Journal
ISSN Print 1687-398X
ISSN Online 1687-3998
ISSN CD-ROM 1687-4005

Table of Contents

ICGST International Journal on Graphics, Vision and Image Processing (GVIP)

A publication of the International Congress for global Science and Technology - (ICGST)

ICGST Editor in Chief: Dr. rer. nat. Ashraf Aboshosha

www.icgst.com, www.icgst-amc.com, www.icgst-ees.com

editor@icgst.com

Abstracts

P1151705543,
Author="Tieling Chen",
Title="Detail Preserving Sorted Difference Filter",

Abstract: A detail preserving filter that uses the intensity differences between a pixel and its neighbor pixels to eliminate impulse noises with minor variations is proposed. The absolute values of the intensity differences are sorted into a sequence in ascending order and the value at a specific position is used to determine whether the pixel under processing is an impulse noise or not. The method to find the specific position is provided and its feasibility is discussed. Theoretically impulse noises can be correctly selected when the density of the noises are not heavy. The filter preserves more fine details than the standard median filter in general. In the situation when the distribution of impulse noises has minor variations, the filter works better than the commonly used adaptive median filter and also produces cleaner results.

P1151738585,
author="Vladimir A. Kulyukin",
title="GreedyHaarSpiker: An Algorithm for In Situ Detection of Highway Lane Boundaries with 1D Haar Wavelet Spikes",

Abstract: An algorithm is presented for in situ vision-based detection of highway lane boundaries on a raspberry pi computer coupled to a raspberry pi camera. The raspberry pi unit is placed inside a Jeep Wrangler, next to the windshield, and is powered through a 12V-to-5V car charger. The algorithm, called GreedyHaarSpiker, is based on the detection of 1D Haar Wavelet spikes in 1D Ordered Haar Wavelet Transforms of image rows. To obtain experimental video data for daytime driving, the author drove his Jeep with the installed raspberry pi unit on a sunny day in September 2016 (run 1) and a cloudy day with light rain in November 2016 (run 2) at a speed of 55-60 miles per hour on Route 30, a two-lane Northern Utah highway. To obtain data video data of driving on snowy roads and night driving, the author drove his Jeep on the same highway and at the same speed on a day after a heavy snowfall (run 3) in January 2017 and on the same day after sunset (run 4). Each run was approximately 35 miles long. Each video was partitioned into frames and a sample of 360 x 240 PNG consecutive frames was selected from each captured video. The performance of the algorithm was tested in situ on a raspberry pi 3 model B ARMv8 1GB RAM computer on each of the four frame samples. The algorithm is implemented in Python 2.7.9 with OpenCV 3.0. The current implementation processes 20 frames per second.

P1151714561,
author="Poorva Waingankar and Sangeeta Joshi",
title="Video Compression using Efficient Encoding Techniques for Low Bit Rate Applications",

Abstract: This paper presents use of Accordion technique along with modified Run Length Encoding for video compression, which consists of exploiting the high amount of temporal redundancies present in videos by converting them to spatial redundancy and using 2D DCT. The Video compression steps are either optimized or completely revamped to meet the compression and video quality requirement in mobile application. This technique is less complex to suit lower end CPUs and achieves a very good compression ratio to suit the narrow bandwidth environments of wireless networks, without compromising on the quality of the video.

P1151733581,
author="Ayush Purohit and Shardul Singh Chauhan",
title="A Precise Technique for Hand Gesture Recognition",

Abstract: Vision based methodologies provides a more natural and proficient result when contrasted with traditional strategies which have been utilized for hand gesture recognition. In this paper, we proposed a video based hand gesture recognition. Our approach commences by acquiring the video frame from a source and converting it into 2D binary frame using YCbCr color space. We implemented opening and closing operations to filter the noise from the frame. In order to track and segment the hand gesture we used Kalman filter and convex hull along with convexity defects for detecting hand regions from the frame. Our framework can perceive six kinds of hand gestures at present time.

P1151748594,
author="D. Boopathy and M. Sundaresan",
title="Securing Images on Cloud using Multidimensional Approach",

Abstract: Encryption is one of the methodologies used to maintain and protect the data confidentiality. As per the user data type's requirements, users need to adopt and implement any one of the existing methods. But those encryption methods and standards may not be bound within the user data country regulations, when the users are from different geographical locations. Some of the existing methods are already compromised by hackers and also some of the government agencies are forcing their country based service providers to provide the encrypted information in the name to maintain the country's security. It is very difficult to manage the threats with one method. The proposed method tried its maximum level to reduce the threats by using different points of view. In this proposed method images and the block-based encryption method have been used to protect the normal and sensitive image from the unauthorized access. The proposed method is tested on all proposed encryption types using greyscale in two scenarios. They are Different Images One Type (DIOT) and Single Image All Types (SIAT). The results of the proposed methods are evaluated using PSNR, MSE, Size of the Image and Histogram to verify the image's integrity.

P1151747592,
author="M. Anyayahan and M. Balinas and A. La Madrid and M. Laurel and C. Lopez and R. Tolentino",
title="Rotational invariant Real Time Text Recognition",

Abstract: In everyday life, people always encounter different text images. These text images are in a style of linear or multi-oriented texts in either printed or written form. Due to different orientations of texts in an image, it is a challenge in Optical Character Recognition to recognize this kind of text. In this paper, real time recognition of text in different rotational variations is presented. The performance is done from acquisition of image by a camera and processed by Microsoft Visual Studio. The detection and recognition of text with different rotational variations are achieved by detecting and computing the direction and angle of tilt respectively through the use of geometric and trigonometric principles then recognized by Tesseract optical character recognition engine after counter rotation.

www.icgst.com

Detail Preserving Sorted Difference Filter

Tieling Chen
Department of Mathematical Sciences, University of South Carolina Aiken
471 University Parkway, Aiken, SC 29803, USA
tielingc@usca.edu,

Abstract

A detail preserving filter that uses the intensity differences between a pixel and its neighbor pixels to eliminate impulse noises with minor variations is proposed. The absolute values of the intensity differences are sorted into a sequence in ascending order and the value at a specific position is used to determine whether the pixel under processing is an impulse noise or not. The method to find the specific position is provided and its feasibility is discussed. Theoretically impulse noises can be correctly selected when the density of the noises are not heavy. The filter preserves more fine details than the standard median filter in general. In the situation when the distribution of impulse noises has minor variations, the filter works better than the commonly used adaptive median filter and also produces cleaner results.

Keywords: *Sorted difference filter, median filter, adaptive median filter, impulse noises, detail preserving filter.*

1 Introduction

The standard median filter replaces the intensity value of a pixel being processed with the median value of the intensities in its neighborhood. The filter works very well on eliminating impulse noises, such as salt-and-pepper noises, when the density of the noises is not heavy ([1], [3], [11]). Impulse noises usually have a unipolar or bipolar distribution of intensities at one end or the two ends of the intensity range. In an image corrupted by impulse noises, white dots appearing in dark regions are called salts and dark dots appearing in bright regions are called peppers. There are quite a few impulse noise models adopted in research ([5]), among which the one with the following probability density function is widely used.

$$p(z) = \begin{cases} P_1, & \text{pepper; } z = 0, \\ P_2, & \text{salt; } z = L-1, \\ 1 - P_1 - P_2, & \text{noise free,} \end{cases} \quad (1)$$

where the range of intensities is the interval $[0, L-1]$, and P_1 and P_2 are the corresponding probabilities for peppers and salts, usually called densities. When $P_1 = P_2$, the noise is called a salt-and-pepper noise, and usually $P = P_1 + P_2$ refers to its total density. Because the intensities of peppers and salts are on the two ends of the intensity range of a corrupted image, they can be removed by the standard median filter effectively when P is low. Experimentally, when $P \leq 20\%$, the performance of the standard median filter is perceptually satisfactory.

In fact, the standard median filter still works well even though the noises are not perfect impulses, which display minor variations around the impulses in the distributions. The probability density function of these type of noises can be expressed as

$$p(z) = \begin{cases} p_1(z), & \text{pepper; } z \in [0, \epsilon_1], \\ p_2(z), & \text{salt; } z \in [L-1-\epsilon_2, L-1], \\ p_3(z), & \text{noise free,} \end{cases} \quad (2)$$

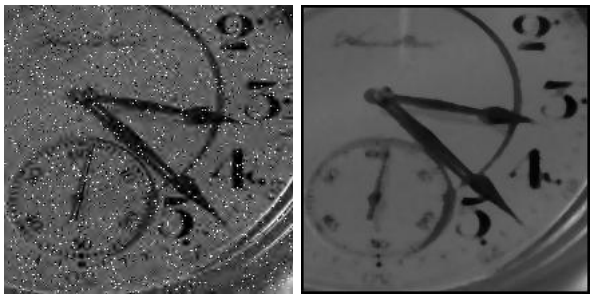
where the small tolerances ϵ_1 and ϵ_2 give two narrow intervals at the two ends of the intensity range $[0, L-1]$, in which noises occur with probabilities described by the functions $p_1(z)$ and $p_2(z)$, and $p_3(z) = 1 - p_1(z) - p_2(z)$ is the probability that a pixel with the intensity z is noise free. Denote

$$P = \int_0^{\epsilon_1} p_1(z)dz + \int_{L-1-\epsilon_2}^{L-1} p_2(z)dz$$

the total density of the impulse noise with minor variations. The shapes of the functions $p_1(z)$ and $p_2(z)$ are not the main concern of the performance of the standard median filter if ϵ_1 and ϵ_2 are relatively small.

However, the standard median filter is not detail preserving, with disadvantages including signal weakening and non-noisy image pixel corruption ([6]). This is because the intensity of a pixel usually is not exactly the median value in its neighborhood covered by the filter mask and it is altered during the process. More specifically, tiny components of objects in an image are usually eroded by the filter. For example, Figure

1 shows an image corrupted by salt-and-pepper noises with a total density $P = 10$ and the result processed by the standard median filter with size 5 by 5. In the resultant image, although the noises are removed, the fine details are heavily eroded and blurred by the filter.



(a) (b)

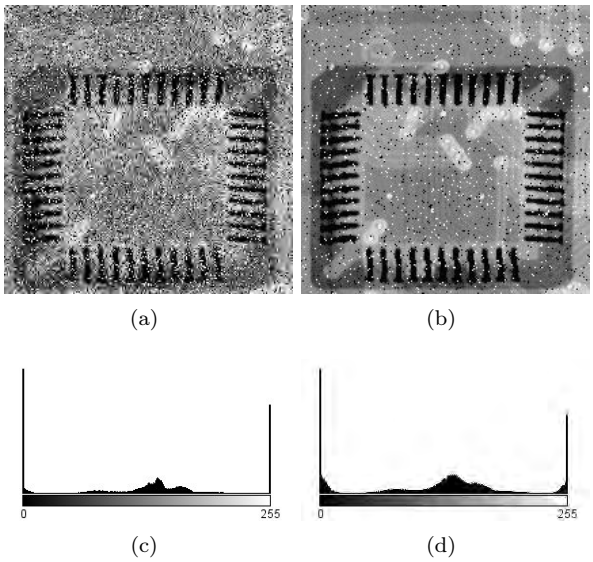
Figure 1: (a) A test image corrupted by salt-and-pepper noises with density 10%. (b) The result by the median filter with size 5 by 5.

Adaptive median filters are more efficient in detail preserving than the standard median filter on images with perfect impulse noises ([2], [3], [4], and [9]). With these filters each pixel is examined and classified as either a noise or a noise free pixel. A designated noise is then processed with the standard median filter or a modified one. For examples, a truncated median filter is used to process designated noises in ([7]); a decision based filter using multiple thresholds with multiple neighborhood information of the center pixel in the filter window is proposed to restore images corrupted by salt and pepper impulse noise ([8]); with modified median filters, decision making filtering technique can be used together with adaptive filters to improve efficiency ([10]).

The essence of the design of an adaptive median filter is on improving the accuracy of noise identification. We use the adaptive median filter in [3] as an example to explain the concept. The filter changes the size of its mask to find a proper median intensity value in a neighborhood of a pixel. If in the same neighborhood the intensity value of the pixel being processed is not an extreme value, its intensity value is not changed, and otherwise it is replaced with the median intensity value. The method tries not to replace the intensity value of a pixel unless it has to do so, in which case either the pixel has an extreme value, which is a candidate of an impulse noise, or the filter reaches its maximum size and the standard median filter must be used.

Generally, adaptive median filters work with the assumption that noises are impulses without variations, usually taking the two extreme intensity values 0 and $L - 1$ in the intensity range $[0, L - 1]$. Unfortunately, the perfect pattern does not always present in applications. In many situations the distribution of the noise intensities shows minor variations around the impulses, which frequently occur when the images are stored with compression, such as in JPEG format. In these situations adaptive median filters may not remove all the noises.

For example, Figure 2 shows an image corrupted by salt-and-pepper noises with minor variations on the impulses and the result obtained by the Adaptive median filter with maximum size 5 x 5. The test image was chopped from a test image used in [3], originally saved in TIFF format but was converted to JPEG format. In the TIFF format, the intensity distribution of the image displays a perfect pattern of two impulses at the two ends, while in the JPEG format, it shows two bumps at the two ends of the distribution, implying the noises are not perfect impulses. In the JPEG image the noises cannot be completely removed by the adaptive median filter.



(a) (b) (c) (d)

Figure 2: (a) An image corrupted by salt-and-pepper noises. (b) The result of the adaptive median filter with maximum size of 5×5 on the image in JPEG format. Some noises still remain in the image. (c) The intensity distribution of the image in TIFF format. (d) The intensity distribution of the image in JPEG format.

Our motivation is to design a filter that efficiently removes impulse noises with minor variations while keeping the non-noisy pixels unaltered. When impulse noises have minor variations, adaptive median filters can not remove the noises cleanly and the standard median filter ruins the fine details. The proposed filter introduced in the next section classifies noises and non-noisy pixels with high accuracy even though the impulse noises have minor variations and therefore details can be well preserved.

The remainder of the paper is organized as follows: Section (2) introduces the new filter and analyze the mathematical mechanism behind it. Section

(3) demonstrates the effectiveness of the new filter by comparing with the standard median filter and the adaptive median filter used in [3]. Section (4) is the summary and conclusions.

2 Sorted difference filter

For a pixel (x, y) in a given image, denote

$I(x, y)$ the intensity at the pixel (x, y), and
S_{xy} a neighborhood centered at (x, y).

Suppose S_{xy} is encompassed by a filter centered at (x, y) with a size $m \times n$. For every pixel $(x', y') \in S_{xy}$ we compute the absolute value of the intensity difference between the pixel (x', y') and the center pixel (x, y),

$$d_{x'y'} = |I(x, y) - I(x', y')|,$$

and call it the *absolute difference* for the pixel (x', y'). Denote D_{xy} the sorted sequence of all the absolute differences found in S_{xy} in ascending order,

$$D_{xy} = \text{ the sorted sequence of } \{d_{x'y'} | (x', y') \in S_{xy}\}.$$

Let l be the length of the sorted sequence D_{xy}, then $l = mn$. We use an index i to access a particular value in D_{xy}, then

$$D_{xy} = \{D_{xy}[i]\}_{i=0}^{l-1}.$$

It is easy to see that $D_{xy}[0] = 0$ and $D_{xy}[i]$ is non-decreasing in i.

We use a threshold T on a specific value $D_{xy}[i^*]$, $0 \le i^* < l$, to select noise candidates. The threshold T is an experimental value that can be adjusted in applications, while the index i^* is mainly determined by the size of the filter and the density of the impulse noises in the image.

The threshold T is selected in such a way that noises with absolute differences higher than T can be selected out. To determine i^*, suppose the densities of salts and peppers are both p, then empirically there are about pl salts and pl peppers in S_{xy}. Let i^* be the nearest whole number no less than pl, giving by the ceiling function

$$i^* = \mathit{ceiling}(pl). \tag{3}$$

The index i^* is the property of the filter and it applies to all the pixels under processing. Once i^* is determined, it does not change during the processing.

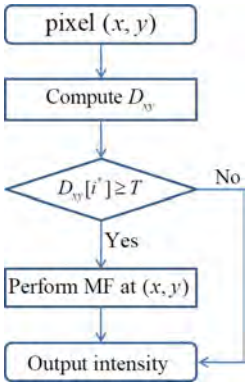
If (x, y) is a noise, say a salt, because we expect there are i^* salts in S_{xy}, then for $0 \le i < i^*$, $D_{xy}[i] = 0$ if the impulse distribution has no variations, or $D_{xy}[i] < T$ for a proper threshold T if the impulse distribution has minor variations. The value $D_{xy}[i^*]$ is the first absolute difference between the salt noise and the background with an abrupt increment. If $D_{xy}[i^*] \ge T$ then the salt noise can be singled out. The situation is the same when (x, y) is a pepper.

For every pixel (x, y), let n_{xy} be the number of pixels in S_{xy} such that the absolute differences are less than the threshold T. Obviously $D_{xy}[i] < T$ for $0 \le i < n_{xy}$ and $D_{xy}[n_{xy}] \ge T$. From the above analysis, if (x, y) is a noise, then statistically $n_{xy} \le i^*$. If (x, y) is a non-noisy pixel and $n_{xy} > i^*$ then the pixel can be easily differentiated from a noise.

The number n_{xy} changes with the pixel location. The filter is based on that the following condition can be reached for a general non-noisy pixel:

$$n_{xy} > i^*. \tag{4}$$

We will see that the condition is generally satisfied if the noise density is not too heavy. For every pixel (x, y) find the sequence D_{xy} in its neighborhood S_{xy} encompassed by the filter. If $D_{xy}[i^*] \ge T$ then (x, y) is regarded as a noise and its value is replaced by the median intensity value in S_{xy}; otherwise, the pixel is treated as a non-noise pixel and its value is unchanged.



The objective of the method is to apply the standard median filter only for the impulse noises and keep the non-noisy pixels unaltered. The effect of the method is determined by the correctness of classifying noises and non-noisy pixels.

For a non-noisy pixel (x, y), there are a few factors affecting n_{xy}. If the absolute differences of all non-noisy pixels in S_{xy} are less than T and the absolute differences of all noises in S_{xy} are not less than T, which usually occurs when (x, y) is in a region with slow intensity changes, then n_{xy} is approximately equal to $l - 2pl$ because there are about $2pl$ noises in S_{xy}. We use the closest whole number that is no more than $l - 2pl$ for n_{xy} in this case,

$$n_{xy} = \mathit{floor}(l - 2pl). \tag{5}$$

For condition (4) to be satisfied, there should be

$$l - 2pl > pl, \tag{6}$$

which yields $p < 1/3$. This means when the impulse noise density is not heavy, a non-noisy pixel can be

correctly identified in general. The filter is equivalent to the standard median filter if the densities of both salts and peppers are not less than 1/3, or the overall density of the impulses is not less than 2/3.

Notice that the non-noisy pixel (x, y) could be on an edge in the image. If (x, y) is on the boundary between two regions with certain sizes and the filter size is relatively small so the boundary is approximately straight, then generally there are at least half of the non-noisy pixels in the same region of (x, y). Suppose the fraction of such pixels in the same region is q and the intensity difference between the two regions is not less than T, then $n_{xy} = (l - 2pl)q$ approximately. We use

$$n_{xy} = \mathit{floor}\left((l - 2pq)q\right). \tag{7}$$

To keep the edge sharp, the pixel (x, y) should not be altered. This requires that the inequality (4) holds, which is

$$(l - 2pl)q > pl, \tag{8}$$

giving

$$p < q/(1 + 2q). \tag{9}$$

Using the empirical value $q \approx 0.5$, we get $p < 0.25$. As mentioned in [3], the standard median filter performs well when $p < 0.2$. Under the same condition, the new filter works better on edge preserving than the standard median filter because edge pixels are not altered.

Theoretically, if the threshold T is properly selected and the densities of the salts and peppers are not heavy, for example if they are less than 0.25, the noises and the non-noisy pixels can be recognized correctly by the new filter. The filter checks $D_{xy}[i^*] \geq T$ to determine whether a standard median filter should be used at (x, y). The threshold T can be adjusted in processing to get an optimal result. When $T = 0$ then $D_{xy}[i^*] \geq T$ is always true so the new filter is exactly the same as the standard median filter. When T is large enough such that $D_{xy}[i^*] \geq T$ can never be satisfied then the new filter does not change anything of the image.

Practically, even though the density p is known beforehand the index i^* may not work well to remove all the noises. This is because digital images are discretely represented and noises are not perfectly evenly distributed, and therefore noises may form clusters with sizes larger than i^*. If a noise (x, y) is in such a cluster and there are more than i^* pixels of the cluster in S_{xy}, then $D_{xy}[i^*]$ is very small and it cannot be detected by the threshold. To remove noise clusters the threshold should be applied to $D_{xy}[j]$ with a bigger index $j > i^*$. Such an index j must be no bigger than n_{xy}, otherwise non-noisy pixels cannot be correctly recognized.

Generally, when the noise density is not heavy there is a big gap between i^* and n_{xy} for every non-noisy pixel (x, y). This means a proper index j can be easily selected such that $i^* < j \leq n_{xy}$. For example, when $p = 0.1$ and the size of the filter is 5×5, which gives $l = 25$, then by (3) $i^* = 3$. To compute n_{xy}, we choose (7) with $q = 0.5$ because the resultant value is smaller than the value given by (5), and get $n_{xy} = (l - 2pl)q = 10$. Then j can be selected in the range $3 < j \leq 10$. The gap between i^* and n_{xy} provides flexibility for the filter to find a proper index j in applications without knowing the actual density of the impulses. Notice that a big j also remove some tiny components of objects in an image.

In summary, for a given image corrupted by impulses with minor variation, select a filter size $l = m \times n$ and set up initial values of the index j and the threshold T. If the impulse density p can be obtained then j can be initially chosen from $lp < j < (l - 2pl)q$ with $q = 0.5$; if not, j can still be easily selected because of the big gap between i^* and n_{xy}. For every pixel (x, y) under processing, sort the absolute differences of pixels in the neighborhood encompassed by the filter in ascending order to get the sequence D_{xy}. If $D_{xy}[j] \geq T$, then the standard median filter with the same filter size is applied; otherwise, the intensity value at (x, y) is unchanged. Finish processing the image to get a result. Adjust the threshold T with each j, and also adjust j and the size of the filter as needed until an optimal result is obtained.

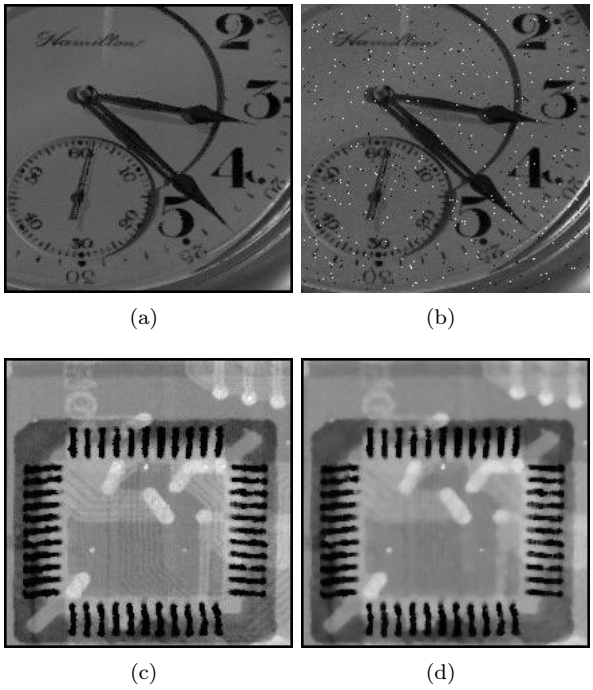
3 Experimental results

In our experiments, the new filter is compared with the standard median filter and the adaptive median filter in ([3]).

The top two images in Figure 3 (a) and (b) display the results of the new filter and the adaptive median filter on the test image in Figure 1 (a), respectively. The image in Figure 3 (a) is obtained by the new filter with a size 5×5, $j = 8$, and $T = 40$. Compared with the image in Figure 1 (b) obtained by the standard median filter, the new result is much perceptually better because it keeps more details of the watch. The Peak Signal to Noise Rate (PSNR) of the image in Figure 3 (a) is 30.31, which is apparently higher than the PSNR of the image in Figure 1 (b), 27.76. Notice that RSNR is only a coarse estimate of the effectiveness of a denoising method, and the perceptual effect of the method is also an important consideration. The image in Figure 3 (b) is the result obtained by the adaptive median filter. Because the impulse noises have minor variations some noises are not cleanly removed. The PSNR of this resultant image is 23.86 owing to the remaining peppers and salts.

At the bottom of Figure 3, results with two different methods on the test image in Figure (2) (a) are displayed. With the new filter, by setting the filter size 5×5, $j = 10$ and threshold $T = 22$, we get the result shown in Figure 3 (c). For comparison, the result


obtained by the standard median filter with the same size is also displayed in Figure 3 (d). Perceptually, the fine details are kept very well in the image obtained by the new filter but they are severely blurred by the standard median filter. The PSNRs are 25.31 for image in Figure 3 (c) and 24.74 for the image in Figure 3 (d), with the new filter giving the higher one. Because the impulse noises have variations on their intensities, some noises cannot be removed by the adaptive median filter, shown in Figure 2 (b), which is not acceptable because of the remaining noises.



(a) (b)

(c) (d)

Figure 3: (a) The result obtained by the new filter on the test image in Figure 1. (b) The result obtained by the adaptive median filter. (c) The result of the new filter on the test image in Figure 2 (a). (d) The result by the standard median filter with size 5×5.

Next, we display the comparisons on the commonly used test images Barbara, Boat, Cameraman and Lena. For each test image, we added impulsive salt and pepper noises by 10%, 20%, 30% and 40%, respectively, and then converted the noisy images into JPEG format. Then the adaptive median filter (AMF), the standard median filter (MF) with size 5×5, and the new filter (NF) with size 5×5, a proper index j and a threshold T were applied on these images. Images with 20% salt and pepper noises and the corresponding filtered images by the mentioned filters are shown in Figures 5 through 8. All PSNR comparisons are summarized in a table at the end. Because the test images with noises are saved in JPEG format, the noise distributions all have minor variations, shown at the two ends of the intensity range of each distribution chart in Figure 4.

Figure 5 displays the comparisons on the noise corrupted image Barbara in JPEG format. The top left image is the test image corrupted with 20% salt and pepper noise and the top right is the result by NF. The bottom left is the result by AF and the bottom right is the result by MF. Among the three filtered images, the image by NF shows the best perceptual effect.

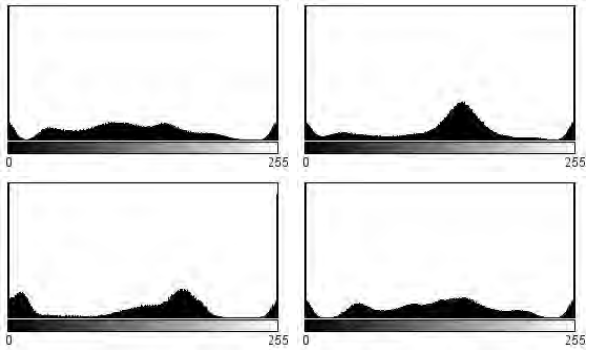


Figure 4: Intensity distributions of the test images corrupted with 20% salt and pepper noise, saved in JPEG format. Top left for Barbara; Top right for Boat; Bottom left for Cameraman; Bottom right for Lena.

Figure 5: Top left, the noise corrupted image. Top right, the result by the new filter. Bottom left, the result by the adaptive median filter. Bottom right, the result by the standard median filter.

Figures 6, 7 and 8 display similar comparisons for the noise corrupted test images Boat, Cameraman and Lena.

Among the above images, the results obtained by the new filter preserve more fine details than the results obtained by the standard median filter. The results obtained by the adaptive median filter still contain noticeable noises so they are not of satisfactory.

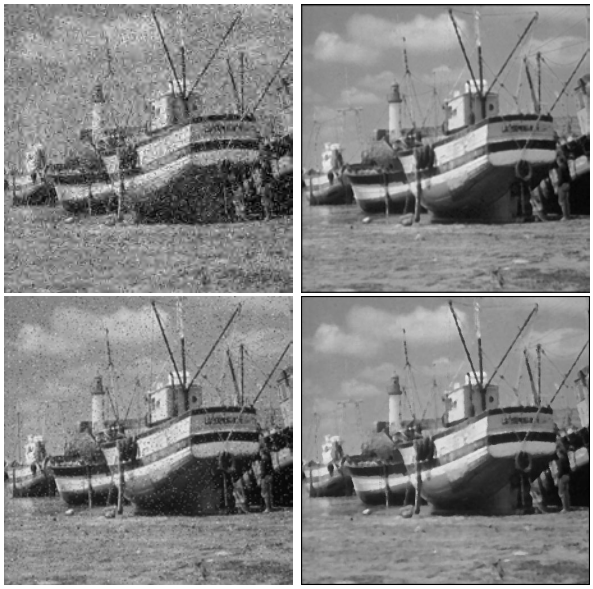


Figure 6: Top left, the noise corrupted image. Top right, the result by the new filter. Bottom left, the result by the adaptive median filter. Bottom right, the result by the standard median filter.

Figure 7: Top left, the noise corrupted image. Top right, the result by the new filter. Bottom left, the result by the adaptive median filter. Bottom right, the result by the standard median filter.

The PSNRs of the filtered images for the test images corrupted with different percentages of salt and pepper noises are listed in the following table. For a test image corrupted with noise less than 40%, the PSNR of the result obtained by the new filter is the highest. When the noise percentage is higher than 40%, the effect of the new filter is essentially the same as that of the standard median filter.

Figure 8: Top left, the noise corrupted image. Top right, the result by the new filter. Bottom left, the result by the adaptive median filter. Bottom right, the result by the standard median filter.

Image	Noise percentage	PSNR for different filters		
		AMF	MF	NF
Barbara	10%	22.75	23.04	24.02
	20%	18.93	22.88	23.02
	30%	15.75	22.65	22.68
	40%	13.72	22.15	22.15
Boat	10%	23.78	26.87	27.86
	20%	19.35	26.58	27.24
	30%	15.95	25.90	26.19
	40%	13.93	25.04	25.10
Cameraman	10%	23.12	23.50	24.65
	20%	18.80	23.29	23.71
	30%	15.38	22.77	22.83
	40%	13.31	22.11	22.05
Lena	10%	24.06	30.33	32.00
	20%	19.43	29.90	30.57
	30%	16.02	29.06	29.22
	40%	13.95	27.75	27.76

4 Conclusion

A novel method is proposed to eliminate impulse noises with minor variations. The method uses a filter to find out impulse noises and replace them with the median intensity values in their neighborhoods, while the non-noisy pixels are not altered. The proposed filter outperforms the standard median filter in fine details preserving because only the noises are processed. The new filter is immune to minor variations of the impulses, so it is more applicable than the adaptive median filter that works well on details preserving and noises elimination only when the impulses have no variations. Theoretically, when the density of the impulses is not heavy the noises can be correctly

identified. The parameters used in the method can also be easily adjusted in applications to obtain optimal results.

References

[1] Arce, G. R., Nonlinear Signal Processing: A Statistical Approach, Wiley: New Jersey, USA, 2005

[2] Eng, H. L., and Ma, K. K., Noise adaptive soft-switching median filter, IEEE Transactions on Image Process., vol. 10, no. 2, pp. 242âĂŞ251, February 2001

[3] Gonzalez, R. C., and Woods, R. E., Digital Image Processing, 3rd edition, Pearson Prentice Hall, New Jersey, USA, 2008

[4] Hwang, H. and Haddad, R., Adaptive median filters: new algorithms and results, IEEE Transactions on Image Processing, vol.4, no.4, pp. 499-502, April 1995

[5] Ibrahim, H., et al., Impulse Noise Model and Its Variations, International Journal of Computer and Electrical Engineering, vol. 4, no. 5, pp. 647-650, October 2012

[6] Judith, G. M. C. and Kumarasabapathy, N. Study and Analysis of Impulse Noise Reduction Filters, Signal & Image Processing: An International Journal(SIPIJ), Vol.2, No.1, March, 2011

[7] Kulkarni, R. N. and Bhaskar, P. C., Decision Based Median Filter Algorithm Using Resource Optimized FPGA to Extract Impulse Noise, Journal of Embedded Systems, vol. 2, no. 1, pp. 18-22, 2014

[8] Lin T. and Yu P., Salt-Pepper Impulse Noise Detection and Removal Using Multiple Thresholds for Image Restoration, Journal of Information Science and Engineering, vol. 22, pp. 189-198, 2006

[9] Meher, S. K., Singhawat, B., An improved recursive and adaptive median filter for high density impulse noise, AEU - International Journal of Electronics and Communications, vol 68, no. 12, pp. 1173-1179, December 2014

[10] Suman, S., Image denoising using new adaptive based median filter, Signal & Image Processing : An International Journal, vol.5, no.4, pp. 1-13, August 2014

[11] Vaseghi, S. V., Advanced Digital Signal Processing and Noise Reduction, Second Edition, John Wiley & Sons Ltd, 2000

Biographies

Tieling Chen is currently Professor at University of South Carolina Aiken, USA. He received his PhD in Mathematics and M.Sc in Computer Science from University of Western Ontario, Canada. His main research interests are Image processing, and Color models and applications.

www.icgst.com

GreedyHaarSpiker: An Algorithm for In Situ Detection of Highway Lane Boundaries with 1D Haar Wavelet Spikes

Vladimir A. Kulyukin
Department of Computer Science, Utah State University, 4205 Old Main Hill, Logan, UT 84322, USA
vladimir.kulyukin@aggiemail.usu.edu,
http://www.cs.usu.edu/people/VladimirKulyukin

Abstract

An algorithm is presented for in situ vision-based detection of highway lane boundaries on a raspberry pi computer with a raspberry pi camera. The raspberry pi unit is placed inside a Jeep Wrangler, next to the windshield, and is powered through a 12V-to-5V car charger. The algorithm, called *GreedyHaarSpiker*, is based on the detection of 1D Haar Wavelet spikes in 1D Ordered Haar Wavelet Transforms of image rows. To obtain experimental video data for daytime driving, the author drove a 2016 Jeep Wrangler with the installed raspberry pi unit on a sunny day in September 2016 (run 1) and a cloudy day with light rain in November 2016 (run 2) at a speed of 55-60 miles per hour on Route 30, a two-lane Northern Utah highway. To obtain video data of snowy roads and night driving, the author drove the same vehicle on the same highway and at the same speed on a day after a heavy snowfall (run 3) in January 2017 and on the same day after sunset (run 4). Each run was approximately 35 miles long. Each video was segmented into 360 x 240 PNG frames. A sample of 1,000 consecutive frames was selected from each video. The performance of the algorithm was tested on a raspberry pi 3 model B ARMv8 1GB RAM computer on each of the four frame samples. The algorithm is implemented in Python 2.7.9 with OpenCV 3.0. The current implementation processes 20 frames per second.

Keywords: *Computer Vision, Wavelets, Lane Detection, Autonomous Vehicles*

Nomenclature: *CV - Computer Vision, AV - Autonomous Vehicle, HWT - Haar Wavelet Transform*

1 Introduction

Autonomous vehicles (AVs), i.e., vehicles capable of navigating various environments without human input, have featured prominently in many research and commercial projects for several decades. The CMU Navigation Laboratory (Navlab) has built a series of robot cars, SUVs, and buses since 1984. The latest robotic car, Navlab 11, is a robot Jeep Wrangler equipped with a range of sensors for obstacle avoidance, path planning and following, and pedestrian detection [1]. The European Technology Platform on Smart Systems Integration project has reported significant contributions to collision avoidance, fleet management, autonomous cruise control, and cooperative driving [2]. Over the past several years, both Google and Tesla have been commercializing their self-driving platforms [3, 4].

Proponents of AVs argue that the major benefits of driverless cars include less traffic congestion, enhanced mobility for the elderly and the disabled, significant increases in roadway capacity, and reduction in traffic accidents [5, 6]. The claim about the reduction in traffic accidents is typically supported by the argument that since all driverless cars will use the same algorithms, they will act predictably and in unison with respect to each other.

Opponents of driverless cars argue that the widespread adoption of AVs will result not only in major losses of driving jobs, but also will likely lead to loss of privacy and increased risks of hacking attacks and terrorism [7]. Some researchers argue that lack of stress during driving and more productive time on the road may create additional incentives to live even further from cities, which will increase the carbon footprint of motor transportation systems [8].

While we believe that completely autonomous cars may become a reality in the long term, provided that not only technical failures [9, 10] but also social and legal implications [11] of AV adoption are properly addressed, human drivers are, and will remain indispensable in the short and medium terms. Consequently, it is important to seek solutions that enhance their safety. Robust vision-based lane detection is one such enhancement. Specifically, vision-based lane detection modules may gradually become an integral part of autopilots in semi-trucks to improve the drivers' safety

on long, monotonous highway stretches with low or no traffic. Such autopilots will be similar to the ones already in use in aircraft and ships and will keep the human in the loop in that the decision to engage or disengage the autopilot will be under the driver's control.

In this article, an algorithm, called *GreedyHaarSpiker*, is presented for in situ vision-based detection of marked highway lane boundaries on a raspberry pi computer with a raspberry pi camera board. It is assumed that the lane boundaries are marked with white or yellow lines, as is the case on highways in many countries. The computer-camera unit is placed inside a 2016 Jeep Wrangler, next to the windshield, and is powered through a 12V-to-5V car charger. The algorithm is based on the detection of 1D Haar Wavelet spikes [12] in 1D Ordered Haar Wavelet Transforms (HWT) of image rows. The algorithm is implemented in Python 2.7.9 and OpenCV 3.

This article is organized as follows. In Section 2, related work is reviewed. In Section 3, the concept of a 1D Haar Wavelet Spike (1D HWS) is formally developed. Section 4 describes the proposed algorithm and analyzes its pseudocode. In Section 5, the highway experiments are described and analyzed. Findings and conclusions are presented in Section 6.

2 Related Work

Vision-based lane detection has been the focus of many research and development (R&D) projects in the past two decades. Wang et al. [13] propose a B-Snake based lane detection and tracking model for a range of lane structures. An algorithm, called *CHEVP*, is developed for providing initial positions for the B-Snake model. A minimum error method is proposed to determine the control points of the B-Snake model by the image forces on both sides of a lane. Experimental results suggest that the algorithm is robust against noise, shadows, and illumination variations in captured images of marked and unmarked roads.

Kim [14] presents a lane detection and tracking algorithm to detect lane curvatures, lane changes, and splitting lanes. The detected lane markings are grouped into separate left and right lane-boundary hypotheses to handle merging and splitting lanes. The hypotheses are evaluated and grouped with a probabilistic, Markov-style framework.

Hsiao et al. [15] propose an embedded real-time lane departure warning system (LDWS) for daytime and nighttime driving. The LDWS features a lane detection algorithm based on peak finding for feature extraction to detect lane boundaries. Gaussian smoothing and global edge detection are applied to reduce noise in images. The reported lane detection rates were 99.57% during the day and 98.88% at night on a sample of highway images.

Erickson and Landberg [16] proposed a lane detection algorithm that uses Hough lines [17] combined with a parabolic second degree fitting for curvature detection. On the raspberry pi 2 model, the algorithm's performance was found to be inadequate for high speed driving. However, when the object detection is removed from the algorithm, the algorithm meets the real time performance requirements on the raspberry pi 2 model.

Mandlik and Deshmukh [18] have developed a lane departure detection system that uses the OpenCV library [19] to detect vehicle lane departures on the raspberry pi hardware. The algorithm uses the OpenCV implementations of the Canny Edge detector [20] and the Hough Transform [17] to detect straight and curved lanes. The experiments are conducted on a toy vehicle with a USB camera mounted on top of it for sending images of white paper lanes on a black floor surface to a raspberry pi computer powered by a laptop.

The position advocated in this article is similar to the positions advocated in [16] and [18]: to be economically viable and broadly shareable, vision-based lane detection algorithms must be implemented and tested in situ on off-the-shelf low-voltage hardware platforms such as the raspberry pi. The creation of replicable hardware and software solutions will enable citizen science drivers to build, test, and broadly share replicable driver safety enhancements.

3 Haar Wavelet Spikes

The GreedyHaarSpiker algorithm described in Section 4 depends on the concept of the 1D Haar Wavelet Spike developed in this section. In the 1D Haar Wavelet Transform (1D HWT), a signal is a vector in $R^n, n = 2^k, k \in N$. Following the formalization in [21], let $W_a^{(k)}$ be a $2^k \times 2^k$ matrix for computing k scales of the 1D HWT. This matrix can be effectively computed from the n canonical base vectors of R^n. If $x = (x_0, ..., x_{2^k-1})$ is a signal in R^n, then y is the k-scale 1D HWT of x defined in (1).

$$W_a^{(k)} x^T = y \tag{1}$$

The transform of the signal is given in (2), where $a_0^{(0)} = \mu(y)$ and $c_i^{(j)}$ is the coefficient of the i-th basic Haar wavelet at scale j [22]. For example, (3) defines the matrix for computing the 1D HWT in R^2.

$$y^T = (a_0^{(0)}, c_0^{(0)}, c_0^{(1)}, c_1^{(1)}, ..., c_0^{(k-1)}, ..., c_{2^{k-1}-1}^{(k-1)}) \tag{2}$$

$$W_a^{(2)} = \begin{bmatrix} 0.25 & 0.25 & 0.25 & 0.25 \\ 0.25 & 0.25 & -0.25 & -0.25 \\ 0.50 & -0.50 & 0.00 & 0.00 \\ 0.00 & 0.00 & 0.50 & -0.50 \end{bmatrix} \tag{3}$$

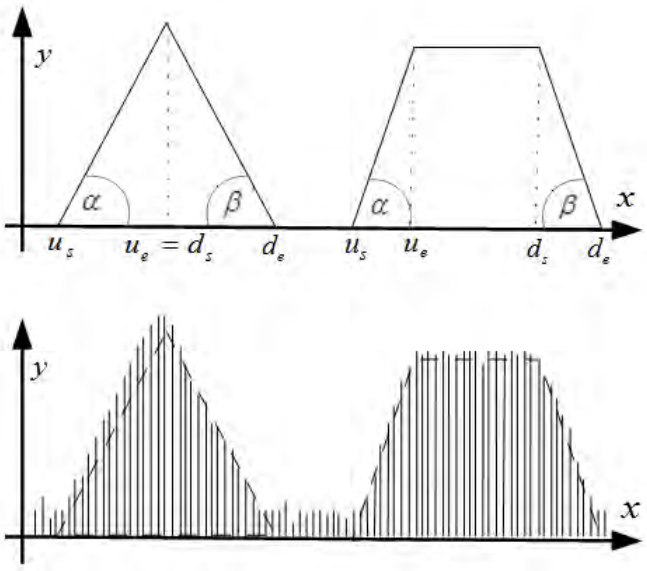



Figure 1: Two types of up-down spikes (above) and the corresponding Haar wavelets at a given scale k from a signal (below).

If the input signal $x = (0, 1, 1, 0)$, then (4) gives the 1D HWT of x computed as $W_a^{(2)} x^T = y$, where $y^T = (0.5, 0, -0.5, 0.5)$.

$$\left[W_a^{(2)}\right] \begin{bmatrix} 0 \\ 1 \\ 1 \\ 0 \end{bmatrix} = \begin{bmatrix} 0.50 \\ 0.00 \\ -0.50 \\ 0.50 \end{bmatrix} \tag{4}$$

It has been theoretically proven that the HWT can detect significant changes in signal values [23]. In this article, we claim that some changes can be characterized as signal spikes [12]. Suppose that there is a finite 1D digital signal. The signal's values may first rise and then fall or they may first fall and then rise. The signal's values may also have a relatively flat segment between the rise and the fall or the fall and the rise. Of course, the signal's values may remain flat for the entire duration of the signal, but a flat signal is not particularly interesting in the sense that it indicates that the underlying phenomenon modeled by the signal does not change.

To model the behavior of 1D digital signals, four types of spikes are postulated: up-down triangle, up-down trapezoid, down-up triangle, and down-up trapezoid. The difference between up-down and down-up spikes is, as their names suggest, the relative positions of the climb and decline segments. In trapezoid spikes, flat segments are always in between the climb and decline segments. One can also view triangle spikes as trapezoid spikes with zero flat segments.

Fig. 1 shows up-down triangle and trapezoid spikes. Fig. 2 shows down-up triangle and down-up trapezoid spikes. In both figures, the lower graphs

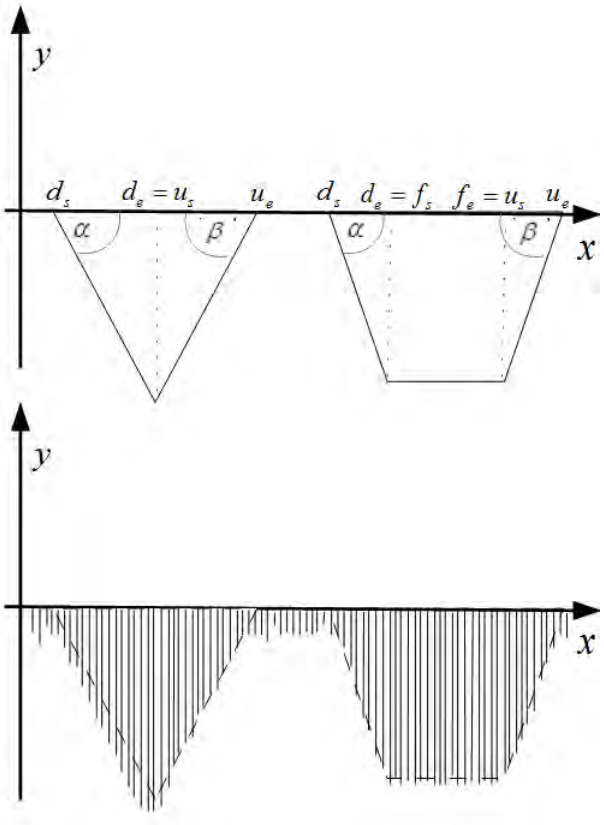


Figure 2: Two types of down-up spikes (above) and the corresponding Haar wavelets at a chosen scale k from a signal (below).

represent the possible values of the corresponding Haar wavelets at a chosen scale k. Up-down spikes describe signals that first increase and then, after an optional flat segment, decrease. Down-up spikes describe signals that first decrease and then, after an optional flat segment, increase.

Let $\mathcal{S}$ be a spike. Then, formally, a spike is a nine element tuple whose elements are real numbers given in (5).

$$\mathcal{S} = (u_s, u_e, \alpha, f_s, f_e, \gamma, d_s, d_e, \beta) \tag{5}$$

The first two elements, u_s and u_e, are the abscissae of the start and end of the spike's climb segment $[u_s, u_e]$, respectively, on which the wavelet coefficients of the 1D HWT increase. If $\omega_{u_s}^{(k)}$ and $\omega_{u_e}^{(k)}$ are the k-th scale wavelet coefficient ordinates at u_s and u_e, respectively, the steepness of the climb, denoted by α, is given in (6).

$$\alpha = tan^{-1}(u_e - u_s, \omega_{u_e}^{(k)} - \omega_{u_s}^{(k)}) \tag{6}$$

As shown in Fig. 1 and Fig. 2, in (5), the flat segments of up-down or down-up spikes, are described by f_s, f_e, and γ, where f_s and f_e in (5) are the abscissae of the start and end of the spike's flat segment, respectively, over which the wavelet coefficients either

Figure 3: A 7-inch raspberry pi touchscreen display where the output of the algorithm is shown. The display is mounted inside a Jeep Wrangler under the rear view mirror in the middle of the windshield.

Figure 4: A raspberry pi v2 camera, shown by a red arrow, is attached to a small cardboard box. The upper edge of the camera is taped to the windshield. The raspberry pi computer, shown by a green arrow, is attached to the back side of the display shown by a blue arrow.

remain at the same ordinate or have minor ordinate fluctuations.

If $\omega_{f_s}^{(k)}$ and $\omega_{f_e}^{(k)}$ are the k-th scale wavelet coefficients corresponding to f_s and f_e, respectively, the spike's flatness, denoted by γ, is defined in (7).

$$\gamma = tan^{-1}(f_e - f_s, \omega_{f_e}^{(k)} - \omega_{f_s}^{(k)}) \tag{7}$$

The numbers of d_s and d_e in (5) are the abscissae of the start and end, respectively, of the spike's decline segment $[d_s, d_e]$, over which the wavelet coefficients of the 1D HWT decrease.

If $\omega_{d_s}^{(k)}$ and $\omega_{d_e}^{(k)}$ are the k-th scale wavelet coefficient ordinates at d_s and d_e, respectively, the steepness of the decline, denoted by β, is given in (8).

$$\beta = tan^{-1}(d_e - d_s, \omega_{d_e}^{(k)} - \omega_{d_s}^{(k)}) \tag{8}$$

Figure 5: The detected lane boundaries are graphically displayed in the bottom right corner of the display. The green arrow points to the detected left boundary. The red arrow points to the detected right lane boundary.

4 GreedyHaarSpiker: Detection of Lane Boundaries

Figures 3, 4, and 5 show the hardware on which *GreedyHaarSpiker*, the lane boundary detection algorithm described in this article, currently runs. In Fig. 3, a seven-inch raspberry pi touchscreen monitor is shown. As shown in Fig. 4, the monitor is attached to a raspberry pi 3 model B ARMv8 computer with 1GB RAM identified by a green arrow. The computer is attached to the back of the monitor and coupled to a raspberry pi camera board v2. The camera, identified by a red arrow, is attached to a small cardboard box and taped to the windshield for balance. In the future, more stable structures will be designed and deployed.

In Fig. 5, the monitor displays the left and right lane boundaries as they are being detected by the algorithm in real time as the vehicle is driven. The whole system is powered through a 12V-to-5V car charger where the USB power line for the raspberry pi is plugged.

Figure 6: Sample frame from video 1.

Algorithm 1 gives the pseudocode of the procedure *detectLaneBoundaries*. The procedure takes as input a 360×240 PNG image like the one shown in Fig. 6. The ROI in the bottom center of the image, shown as a white rectangle in the center of the image in Fig. 7,

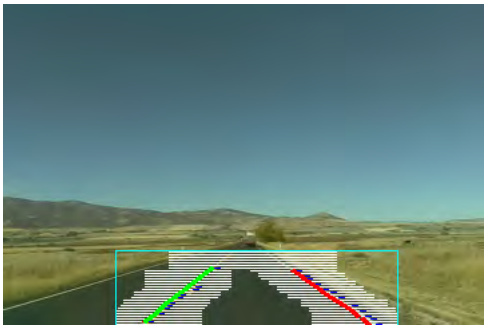


Figure 7: A region of interest with scanlines and detected lane boundaries.

Figure 8: The lane boundaries detected in the frame in Fig. 6. The green line marks the left boundary; the red line marks the right boundary.

is cropped. The cropped ROI is grayscaled, blurred with the 7×7 Gaussian kernel, and thresholded with the Otsu thresholding operator.

The procedure *greedyHaarSpiker*, outlined in Algorithm 2, is applied to the preprocessed ROI. This procedure returns two lists, LPoints and RPoints, of (x, y) tuples. The procedure *fitLine* uses linear regression to fit a line through LPoints and RPoints. The lines are filtered by slope to reduce false positives. The slope thresholds for the left boundary are from -60 to -30; the slope thresholds for the right boundary are from 30 to 60. If the line through LPoints passes the left slope threshold filter, it is taken to be the left boundary of the vehicle's lane. If the line through RPoints passes the right slope treshold filter, it is taken to be the right boundary of the vehicle's lane.

Algorithm 1 *detectLaneBoundaries*(Img)

```
ROI ← cropROI(Img);
ROI ← convertToGrayscale(ROI);
ROI ← gaussianBlur(ROI);
ROI ← thresholdOTSU(ROI);
LPoints, RPoints ← greedyHaarSpiker(ROI);
LeftLaneBoundary ← fitLine(ROI, LPoints);
RightLaneBoundary ← fitLine(ROI, RPoints);
return LeftLaneBoundary, RightLaneBoundary;
```

Algorithm 2 *greedyHaarSpiker*(ROI, s_r, e_r, Δ)

```
LLPoints ← [];
RPoints ← [];
LSpike ← NULL;
RSpike ← NULL;
r ← s_r;
while r ≥ e_r do
  LLine ← getLeftScanLine(ROI, r, LSpike);
  RLine ← getRightScanLine(ROI, r, RSpike);
  LHWT ← ordHWT(LLine);
  RHWT ← ordHWT(RLine);
  LSpike ← detectSpike(LHWT);
  RSpike ← detectSpike(RHWT);
  if LSpike ≠ NULL then
    LPoints.add(LSpike.getMidPointOfClimb());
  end if
  if RSpike ≠ NULL then
    RPoints.add(RSpike.getMidPointOfClimb());
  end if
  r ← r + Δ;
end while
```

The procedure *greedyHaarSpiker* in Algorithm 2 takes a preprocessed ROI and three integer parameters s_r, e_r, and Δ. The parameters s_r and e_r ($s_r \geq e_r$) specify the start and end rows, respectively, in the ROI where the spikes are detected. The parameter Δ specifies a step value, a small negative integer, for generating the exact row numbers where spikes are detected. For example, if the algorithm is to detect spikes in the row range $[50, 40]$, i.e., $s_r = 50$ and $e_r = 40$, with $\Delta = -2$, the sequence of rows that will be considered is $(50, 48, 46, 44, 42, 40)$. Note that the spike detection starts from the lower rows that are closest to the vehicle and moves up to the the rows that are further away from the vehicle.

The variables LPoints and RPoints contain the (x, y) tuples returned to *detectLanes* after linear regression line fitting. The variables LSpike and RSpike contain two spikes detected in the ordered HWTs of the left and right scanlines, respectively.

In the **while**-loop of *greedyHaarSpiker*, two scanlines, LLine and RLine, of 64 pixels each are chosen on the left and right sides of the ROI in row r. The scanline's length, i.e., 64, can be changed through a global variable but it has to be equal to an integral power of 2. A value of 64 was experimentally found to result in optimal performance.

If the value of LSpike is NULL, the left scanline starts at column 0. Similarly, if the value of RSpike is NULL, the right scanline starts at column $w - 1$, where w is the width of the ROI, which, in the current implementation, is equal to 200. If the value of LSpike is not NULL, which means that a spike was detected in the previous row, the left scanline, saved in the LLine variable, is centered on the middle of the two ordinates of the detected spike's climb segment, i.e.,

the ordinates of u_s and u_e in equation 5. The flat and down segments are currently not taken into account in the algorithm. The right scanline is detected and saved in RLine in the same way except that the spike saved in RSpike is used. In Fig. 7, the scanlines are shown as horizontal white lines on the left and right sides of each row. As row number r apporoaches the upper boundary of the ROI (i.e., e_r) the gap between the left and right scan lines becomes smaller.

The procedure *detectSpike* in the **while**-loop of the procedure *greedyHaarSpiker* uses thresholds for the angles of the climb, flat, and decline spike segments, i.e., α, γ, β in (5), and returns the leftmost spike that clears the thresholds. In the current implementation, $\alpha = \beta = 60°$ and $\gamma = 5°$. In other words, the spikes whose climb or decline angles are less than $60°$ are filtered out, and flat segments are detected so long as consecutive wave coefficients fluctuate within $\pm 5°$ of 0. This algorithm is greedy in that it always returns exactly one leftmost spike in each left scanline and exactly one leftmost spike in each right scanline. All other spikes are ignored. If no spikes clear the angle thresholds, the value of NULL is returned.

When **while**-loop of *greedyHaarSpiker* finishes, the lists LPoints and RPoints contain (x, y) tuples representing the mid points of the climb segments of spikes detected in the left and right scanlines in each of the processed rows. These points are used by the procedure *fitLine* in *detectLanes* to fit lines through them. The lines identify the left and right lane boundaries, as shown in Fig. 8.

5 Experiments

The images for the experiments were captured with the hardware shown in Figures 3, 4, and 5. To obtain experimental video data for daytime driving, the author drove a 2016 Jeep Wrangler on two different days in September (run 1) and November 2016 (run 2) at a speed of 55-60 miles per hour on Route 30, a two-lane Northern Utah highway with marked lane boundaries. On each run, the raspberry pi camera unit was turned on to record the video and save it on the raspberry pi's sdcard. The first run was on a sunny day with clear skies and good visibility. The second drive was on a cloudy day with light rain. To obtain experimental video data for driving on snowy roads and night driving, the author drove the same vehicle on the same highway and at the same speed on a day in January 2017 after a heavy snowfall (run 3) and on the same day after sunset (run 4).

The first three runs were approximately 35 miles long. The fourth run was approximately 20 miles long. The video from each run was segmented into frames and a sample of 360 x 240 consecutive PNG frames was selected from it. Samples 1, 2, and 3 had 1,000 frames each selected from the videos recorded in runs 1, 2, and 3, respectively; sample 4, selected from the run 4 video, included 775 frames.

Table 1: Lane boundary detection accuracy

SN	NB = 2 (%)	NB $\geq$ 1 (%)	FP (%)
1	61.90	91.20	1.60
2	34.10	77.40	2.70
3	16.90	64.10	8.30
4	15.74	57.03	11.48

The evaluation of the algorithm's performance was done manually by two human evaluators who compared the lane boundaries drawn in each image by the algorithm with the actual lane boundaries in the same image. Each image thus evaluated was placed into one of the three accuracy categories: both boundaries detected, at least one boundary detected, and no boundary detected. An actual boundary was considered detected accurately if the boundary line drawn by the algorithm was exactly aligned with the actual boundary.

Table 1 shows the accuracy results on all four image samples. The column SN, which stands for sample number, lists the four sample numbers. The second column, NB = 2, shows the percentage of frames where the number of detected boundaries (NB) is exactly 2, i.e., both boundaries are detected. The third column, NB $\geq$ 1, shows the percentage of frames where at least one boundary was accurately detected. The fourth column gives the percentage of false positives (FP) in each sample.

In sample 1, both boundaries were accurately detected in 61.9% of the frames and at least one lane boundary was detected in 91.20% of the images. The percentage of false positives in sample 1 is 1.6%. The detection results in sample 2 were 34.10% for both boundaries and 77.40% for at least one lane boundary. The percentage of false positives in sample 2 was 2.70%. In sample 3, taken on a winter day, the percentage of both lanes detected dropped to 16.90% and the percentage of at least one lane detected decreased to 64.10%. The percentage of false positives in samle 3 increased to 8.30%. Finally, in sample 4, the lane identification performance was the worst. Specifically, the percentage of both lanes detected was 15.74%, the percentage of at least one lane detected was 57.03%, and the percentage of false positives increased to 11.48%.

Fig. 9 illustrates an inaccurate boundary detection and a false positive from sample 1. While the left lane boundary, denoted by a small bright green line, is accurately aligned with a real boundary, it is aligned with the boundary of the opposite lane. This misalignment shows a problem with the greedy approach in that the algorithm always chooses the leftmost spike in each scanline. The red line, almost perpendicular to the bright green line, is a false positive.

Fig. 10 shows two frames from run 1 taken on a sunny day. The left image shows both boundaries detected accurately. The right image shows only the


Figure 9: Run 1: a short green line is inaccurately aligned with a wrong boundary; a red line is a false positive.

Figure 10: Run 1: both lanes recognized (left); only the left boundary recognized (right).

Figure 11: Run 2: both lanes recognized (left); no left boundary recognized and a false right boundary (right).

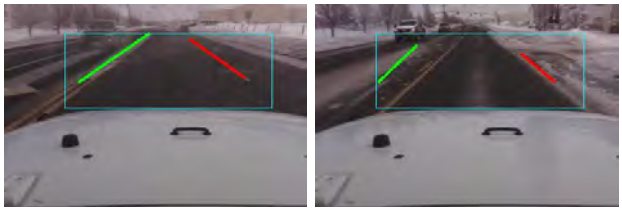


Figure 12: Run 3: both lanes recognized (left); a false left boundary and a correct right boundary (right).

Figure 13: Run 4: both lanes recognized (left); neither lane recognized (right).

left boundary (green line) detected accurately while the right boundary is not detected at all. The failure to detect the right boundary was caused by the shadow of a semitruck in the opposite lane.

Fig. 11 shows two frames from run 2 taken on a cloudy day. The left image shows both lanes accurately recognized. In the right image, the left boundary is not detected and the right boundary is detected inaccurately. This is also a case of a false positive.

Fig. 12 shows two frames from run 3 taken after a heavy snowfall. The left image shows both lanes accurately recognized. In the right image, the left boundary is detected inaccurately and the right lane is accurately recognized. There were many instances of detection failures because the lanes were covered by snow.

Fig. 13 shows two frames from run 4 taken at night after a heavy snowfall. The left image shows both boundaries detected accurately. It is interesting to note that the lane detection recognition was better when there were cars in the opposite lane with their lights turned on. The right image shows a frame where neither boundary was detected.

6 Conclusions

In this article, an algorithm was presented for in situ vision-based detection of highway lane boundaries on a raspberry pi computer coupled to a raspberry pi camera. The algorithm, called *GreedyHaarSpiker*, is based on the detection of 1D Haar Wavelet spikes in 1D Ordered Haar Wavelet Transforms of image rows.

The position advocated in this article is that, in order to be economically viable and broadly shareable, vision-based lane detection algorithms should be implemented and tested in situ on off-the-shelf low-voltage hardware platforms such as the raspberry pi. The creation of replicable hardware and software solutions will enable citizen science drivers to build, test, and broadly share replicable driver safety enhancements.

To address the lane detection problems described in Section 5, several improvements are being considered. Recall that, as explained in Section 4, the flat and down segments are currently not taken into account in the algorithm. Thus, the first improvement


is to use not just the climb segment of each detected spike but also the flat and decline segments when computing the 2D points of a potential line either on the left or on the right. The second improvement is to use a different curve fitting algorithm, e.g., a high order polynomial, to find the best line that fits a set of points. A potential drop in the number of frames processed per second may be compensated by more accurate lane boundary detection. The third improvement is to add geometrical constraints to reduce the number of false positive.

Acknowledgements

The author is grateful to Vikas Reddy Sudini for helping him evaluate the algorithm's performance.

References

[1] S. Thrun. Toward Robotic Cars. *Communications of the ACM*, 53(4):99–106, 2010.

[2] J. Dokic, B. Müller, and G. Meyer. *European Roadmap Smart Systems for Automated Driving.* European Technology Platform on Smart System Integration, Berlin, Germany, 2015.

[3] T. Simonite. Data shows google's robot cars are smoother, safer drivers than you or i. *MIT Technology Review*, Oct., 2013.

[4] G. Nelson. Tesla beams down 'autopilot' mode to model s. *Automative News*, Oct. 14, 2015.

[5] C. Mui. Will the google car force a choice between lives and jobs? *Forbes*, Dec., 2013.

[6] T. Lassa. The beginning of the end of driving. *Motor Trend*, Jan., 2013.

[7] O. Miller. Robotic cars and their new crime paradigms. *LinkedIn Pulse*, Sept. 3, 2013.

[8] M. Ufberg. Whoops: The self-driving tesla may make us love urban sprawl again. *Wired*, Oct. 10, 2015.

[9] D. Yadron and D. Tynan. Tesla driver dies in first fatal crash while using autopilot mode. *The Guardian*, Jul. 1, 2016.

[10] V. Mathur. Google autonomous car experiences another crash. *Government Technology*, Jul. 17, 2015.

[11] J. Boeglin. The costs of self-driving cars: reconciling freedom and privacy with tort liability in autonomous vehicle regulation. *Yale Journal of Law and Technology*, 17(1):Article 4, 2015.

[12] V. Kulyukin and V. R. Sudini. Real-Time Vision-Based Lane Detection on Raspberry Pi with 1D Haar Wavelet Spikes. In *Lecture Notes in Engineering and Computer Science: Proceedings of the International MultiConference of Engineers and Computer Scientists*, pages 75–80, Hong Kong, China, March 2017.

[13] Y. Wang, E. Teoha, and D. Shen. Lane detection and tracking using b-snake. *Image and Vision Computing*, 22:269–280, 2008.

[14] Z. Wang. Robust lane detection and tracking in challenging scenarios. *IEEE Trans. on Intelligent Transportation Systems*, 9(1):16–26, 2008.

[15] P. Hsiao, C. Yeh, S. Huang, and L. Fu. A portable vision-based real-time lane departure warning system: day and night. *IEEE Trans. on Vehicular Technology*, 58(4):2089–2094, 2009.

[16] J. Eriksson and J. Landberg. *Lane departure warning and object detection through sensor fusion of cellphone data.* Master's thesis in Applied Physics and Complex Adaptive Systems, Department of Applied Mechanics, Chalmers University of Technology. Göteborg, Sweden, 2015.

[17] R.O. Duda and P.E. Hart. Use of the hough transformation to detect lines and curves in pictures. *Comm. ACM*, 15:11–15, 1972.

[18] P. Mandlik and A. Deshmukh. Raspberry-pi based real time lane departure warning system using image processing. *International Journal of Engineering Research and Technology*, 5(6):755–762, 2016.

[19] R. Laganiere. *OpenCV 2 Computer Vision Application Programming Cookbook.* Packt Publishing LTD, 2011.

[20] J.F. Canny. A Computational approach to edge detection. *IEEE Trans. on Pat. Anal. And Mach. Intel.*, 8:679–688, 1986.

[21] A. Jensen and A. Cour-Harbo. *Ripples in mathematics: the discrete wavelet transform.* New York: Springer, 2011.

[22] Y. Nievergelt. *Wavelets made easy.* Boston: Birkhäser, 2001.

[23] S. Mallat and W. Hwang. Singularity detection and processing with wavelets. *IEEE Trans. on Information Theory*, 38(2):617–643, 1992.

Biographies

Vladimir A. Kulyukin is an Associate Professor of Computer Science at Utah State University. He holds a Ph.D. in Computer Science from the University of Chicago that he received in 1998. His research interests include AI, computer vision, and sensor fusion.

www.icgst.com

Video Compression using Efficient Encoding Techniques for Low Bit Rate Applications

Poorva Waingankar[1], S.M. Joshi[2]
1.Assoc.Professor, Thakur College of Engg. &Technology, Mumbai, India.
2.Professor, Vidyalankar Institute of Technology, Mumbai, India.
pwaingankar@gmail.com,
http://www.tcetmumbai,in

Abstract
This paper presents use of Accordion technique along with modified Run Length Encoding for video compression, which consists of exploiting the high amount of temporal redundancies present in videos by converting them to spatial redundancy and using 2D DCT. The Video compression steps are either optimized or completely revamped to meet the compression and video quality requirement in mobile application. This technique is less complex to suit lower end CPUs and achieves a very good compression ratio to suit the narrow bandwidth environments of wireless networks, without compromising on the quality of the video.

Keywords:*H.264, Accordion, Run Length Encoding(RLE), Discrete Cosine Transform(DCT), Huffman Encoding*

Nomenclature

DCT	Discrete Cosine Transform
Mbps	Mega Bits per second
MSE	Mean Square Error
PSNR	Peak Signal to Noise Ratio
QP	Quantization Parameter
RLE	Run Length Encoding

1. Introduction
The technological development in multimedia industry over the past decade has enabled widespread usage of internet based applications and smart phones. Out of the various types of media, video transmission and reception through wireless networks is important in the context of the universal access. The increase in communication speed, computing power and availability of computer storage facilities, has led to a new age of multimedia applications. Various applications such as mobile messaging, video conferencing, use of social networking sites etc. require use of multimedia on large scale Although Wireless communications technologies have been evolving rapidly, the available bandwidth is still of great value and so video coding at ultralow bitrates plays an important role in the development of convergent and interoperable video based multimedia services. These applications need storage of high-quality data, reliable transmission and ease of access to content. The volume of data generated by digitizing a video signal is very large for most transmission systems. Therefore, digital video compression is an important aspect in the realization of these applications. The demand for quality, performance and limitations of available transmission capabilities is necessary to be fulfilled by digital video compression techniques. An efficient and well designed video compression system gives significant performance advantages for visual communication at both low and high transmission bandwidths.

The process of transmission and reception of digital video from source to its destination involves many stages. The most important process is compression (encoding) and decompression (decoding). In this the bandwidth-intensive 'raw' digital video is reduced to a manageable size for transmission or storage, then reconstructed for display. The proper compression and decompression process can provide better image quality, greater reliability and/or more flexibility. Therefore, researchers have keen interest in the continuing development and improvement of video compression and decompression methods involving various innovative techniques.

In a typical video, often the temporal redundancies are found to be more relevant than spatial one. In the current video compression techniques, these redundancies are not fully exploited. It is possible to achieve more efficient compression by exploiting these redundancies in the temporal domain. In most of the techniques motion estimation and compensation techniques are usually employed to exploit temporal redundancies. It is observed that the motion estimation process is computationally intensive and its real time implementation is difficult as well. Considering current trends and developments in multimedia applications over internet and mobile communication, an effective algorithm which can fully exploit the redundancy would help to reduce the overall bit rate of transmission/reception.

2. Video Compression Fundamentals

An uncompressed video produces an enormous amount of data and need more than 100s of Mbps bandwidth. Such amount of data causes extremely high computational demands even with powerful computing systems. Hence data compression is an important aspect for managing such data. There are mainly two categories of compression; lossy and lossless. In lossy methods; Transform based coding, Vector quantization, block truncation etc. are used whereas, Run Length Encoding, Huffman Coding, Predictive Coding are used for lossless compression.

The lossless compression retains the original data retaining individual image sequences remain the same, hence compression rate is smaller in this case. The "lossy" compression methods remove image and sound information that is unlikely to be noticed by the viewer, thereby volume of data is significantly decreased. There is always a trade-off between data size and the quality. The higher the compression ratio, lower the size and the quality too. The encoding and decoding process also needs computational resources which need to be taken into consideration. The digital video contains a great deal of redundancy which is categorized in three types as given below:

- Spatial redundancy, which is due to the correlation or dependence between neighbouring pixel values
- Spectral redundancy, which is due to the correlation between different colour planes or spectral bands
- Temporal redundancy, which is present because of correlation between different frames in videos.

The spatial redundancy is reduced by registering differences between parts of a single frame; this is known as intraframe compression and is closely related to image compression. Likewise, temporal redundancy can be reduced by registering differences between frames; this is known as interframe compression, including motion compensation and other techniques. Hence for effective video compression, both interframe and intraframe techniques are used. The typical Video Compression system is shown in Figure.1.

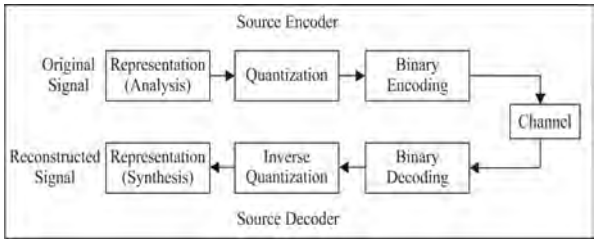


Figure1.Typical Video Compression Scheme

3. Brief Literature Review

The digital video compression technologies have become an integral part of visual information transmitted/received through wired and wireless networks over last one and a half decades. Various standards have been developed for this purpose which define a specific bit stream syntax, imposes very limited constraints on the values of that syntax, and define a limited-scope decoding process. Video codecs are primarily characterized in terms throughput of the channel, distortion of the decoded video, delay and complexity (in terms of computation, memory capacity, and memory access requirements). The intent is for every decoder that conforms to the standard to produce similar output when given a bit stream that conforms to the specified constraints. Thus, these video coding standards are written primarily only to ensure interoperability (and syntax capability), not to ensure quality. This limitation of scope permits maximal freedom to optimize the design of each specific product (balancing compression quality, implementation cost, time to market, etc.). It provides no guarantees of end-to-end reproduction quality, as it allows even crude encoding methods to be considered in conformance with the standard [1][2].

To obtain highly compressed videos without compromising visual quality and to make cost performance trade-offs best suited to applications, researchers have proposed different methods. The multi-objective optimization technique used as a mean for multi-criteria decision making [3]. In which quantization Parameter (QP) controls the tradeoff between quality and bit rate in the sense that a QP increment by 1 results in 12.5% reduction of bit-rate. For network related constraints, optimization algorithm referred to as the Network State Dependent Video Compression Rate (NSDVCR), which determines the compression rates depending on the video characteristics and the network condition is proposed [4].

The possibility of dynamic frame skipping to achieve even higher video compression for low bit rate applications less than 16KbpS is explored by researchers [5]. Motion compensation is very important step in video compression ,so by using control grid interpolation for block based motion compensation, like other interframe compression techniques, produces an approximation of a frame by reusing data contained in the frame's predecessor[6] and in another technique i.e. overlapped block motion compensation is proposed that [7], for each block in the current frame a matching block is found in the past frame and if suitable, its motion vector is substituted for the block during transmission. Depending on the search threshold some blocks will be transmitted in their entirety rather than substituted by motion vectors. The problem of finding the most suitable block in the past frame is known as the block matching problem.

Videos with less motion elements contain high level of temporal redundancy. To avoid the complex computational step of motion estimation and compensation, a new low complexity DCT based video compression method is proposed where Accordion representation converts 3D video content by a 2D image, which allows exploiting the redundancy for high compression [8].

In the subsequent section, use of Accordion representation along with improved Huffman dictionary and modified RLE for Video is presented.

incorporated into the model, it can be shown that significant improvements in the performance of the algorithm can be realised. Moreover, the simplicity and

the efficiency of dynamic pose tracking techniques succeeded to improve the robot pose estimation process.

4. Proposed Methodology

The video signal has high temporal redundancies due to the high correlation between successive frames. It is possible to achieve more efficient compression by exploiting more and more the redundancies in the temporal domain. The proposed method consists of projecting temporal redundancy of each group of pictures into spatial domain to be combined with spatial redundancy in one representation with high spatial correlation i.e. by using **Accordion representation**. The Accordion representation provides a symmetric encoder-decoder design, avoiding the motion compensation step and reduces blocking artifacts. The Accordion representation of any video acts like a preprocessing technique for DCT to achieve a very good amount of energy compaction. The flow chart of an implementation of proposed algorithm is shown in Figure.2.

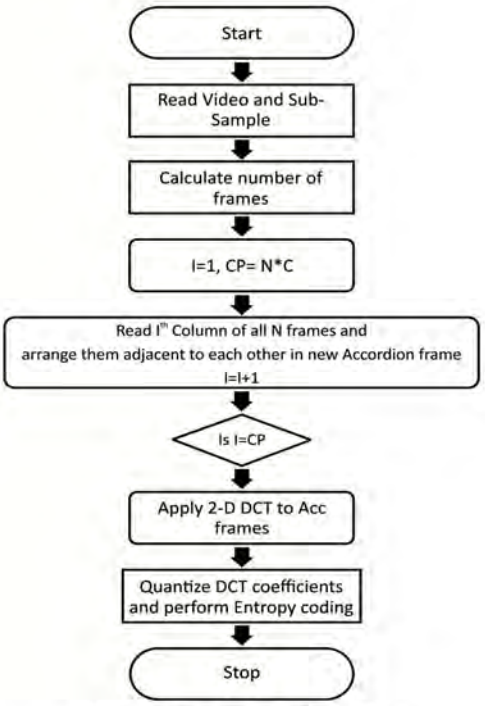


Figure 2: Flow Chart of algorithm

Initially for a video frame, Sub-sampling is implemented by calculating the average pixel value for each group of several pixels, and then substituting this average in the appropriate place in the approximated image. In general, whenever sub sampling is done at the encoder, the decoder has to reconstruct the original picture with some approximation by using a technique called pixel doubling. But in mobile based applications, since the screen resolution is less due to small size, this pixel doubling step is avoided which reduces the decoder complexity.

After considering various factors like the compression percentage, the computational complexity and picture clarity, bilinear interpolation method is found to give the best performance in picture clarity with moderate complexity. After being read into matrix, the input video is divided into several groups with each group consisting of N number of frames where N is the number of frames played in the video per second i.e. fps of the input video. This group having similar temporal frames are gathered into one stretched frame (2 dimensions) by reading each column of every frame subsequently.

The final step consists of coding the obtained frame. The image obtained from the previous steps is now divided into blocks of size 8 x 8, which are then transformed using an 8 x 8 forward DCT. The top-left coefficient in the 2-D DCT array is referred to as the DC coefficient and is proportional to the average brightness of the spatial block. The low-frequency coefficients in the top-left comer of the array have larger values than the higher-frequency coefficients. The transform coefficients are then quantized as per their statistical properties. Most of the energy is concentrated in the low frequency coefficients and hence the higher frequency coefficients which are the least important are harsh quantized or forcibly reduced to zero to avoid any further processing. The Quantization table is designed to provide the most visually correct reconstruction Image. It is designed according to the perceptual importance of the DCT coefficients under the intended viewing conditions. The quality and bit rate of an encoded image can be varied by changing this array. The quantization of AC coefficients creates many zeros, especially at higher frequencies which can be coded efficiently.

The following relation is used for quantization.

$$QDCT = round\ [(8*DCT)/scale*Q) \quad (1)$$

Where, *DCT* is the DCT coefficients, Scale is the scaling factor, Q is the corresponding element of the quantization matrix.

The 2-D array of the DCT coefficients is now formatted into a 1-D vector using a zigzag reordering. Hence the 8 x 8 DCT matrix is now converted to a one dimensional array of 64 coefficients. These 64 numbers are collected by scanning the matrix in zigzag fashion. This rearranges the coefficients in approximately decreasing order of their average energy (as well as in order of increasing spatial frequency) with the aim of creating large runs of zero values since it produces a string of 64 numbers that starts with some non-zeros and typically ends with many consecutive zeros. These runs of zeros are further compressed efficiently using the modified run length encoding procedure.

When the two DC coefficients belonging consecutive DCT matrices have a large difference every such unique difference leads to one unique symbol in the Huffman dictionary in turn leading to many code words which defeats the purpose of compression. To resolve this issue, difference between thee coefficients is coded digit wise with ten unique symbols, thereby code words consequently leading to a much smaller Huffman dictionary. This approach has tremendously reduced the dictionary size and increased the compression ratio.

While carrying out the compression for different videos, it is observed that apart from the number zero, there are very few symbols which have frequent repetitions and hence conventional RLE is not suitable here. This

problem is resolved in the following manner. After analysing the input stream of quantized DCT coefficients in the modified RLE,

a. There is no Run Length Encoding for non-zero elements
b. RLE for all zeros encountered until the last non-zero element
c. Once the last non-zero element is encountered, all the remaining zeros are replaced by special end–of-block (EOB) code with 2 'Zeros'.
Upon reception of the EOB signal, the receiver automatically sets all the remaining coefficients along the zigzag scan to zero. For decoding bit stream, exactly reverse process is carried out step by step. Once the Accordion frame is reconstructed, the MSE and PSNR which are the metrics for reconstructed video quality were calculated using the following relations.

$$PSNR = 10\ log_{10}\ (Max^2/MSE) \quad (2)$$

Where *Max* is the maximum possible intensity in the image (e.g. 255 for a sample precision of 8 bits), and the Mean Square Error (MSE) is given by:

$$MSE = \frac{1}{mn}\sum_{i=0}^{m-1}\sum_{j=0}^{n-1}\left[I(i,j) - K(i,j)\right]^2 \quad (3)$$

Where the number of rows and columns in the image are *m* and *n* respectively.
I(i,j) is the intensity of a pixel at position (i,j) in the original Accordion image, while *K(i,j)* is the value of the corresponding pixel in the compressed and reconstructed Accordion image. The compression in percent is given by;

$$\%C = \frac{(\text{Size of Ori. Video} - \text{Size of Compressed Video})}{\text{Size of Ori. Video}} \quad (4)$$

5.Results

After applying Accordion principle to frames of input video, a stretched frame is formed as shown in Figure 3. This is constructed from 4 sample frames. It can be observed that the temporal redundancies present among the four sample frames is converted to spatial redundancies in the resulting Accordion frame. This step acts as the preprocessing tool to make the 2D DCT very efficient.

Figure 3. Stretched Accordion Frame

After applying 2D-DCT and quantization, it is observed. that the dictionary size reduces to a great extent by using modified RLE and efficient handling of DC coefficients. which is shown in Figure 4. Table 2 shows that for the case of 10 frames, the average length of code words reduces from 3.7375 in conventional technique to 2.877 in improvised technique.

Table 2. Codeword-length with improved RLE and DC

Average Length of Code words					
Number of Frames	2	4	6	8	10
RLE & DC	3.6556	3.6353	3.6893	3.7314	**3.7375**
RLE &Improved DC	3.1162	3.1919	3.1603	3.2349	3.2021
Modified RLE & DC	2.9048	2.8795	2.853	2.8981	**2.8777**

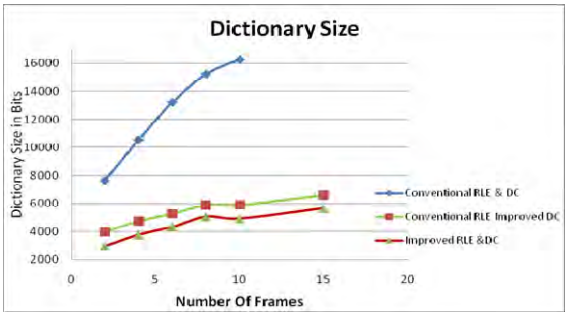


Figure 4. Dictionary Size for RLE and DC Coefficients

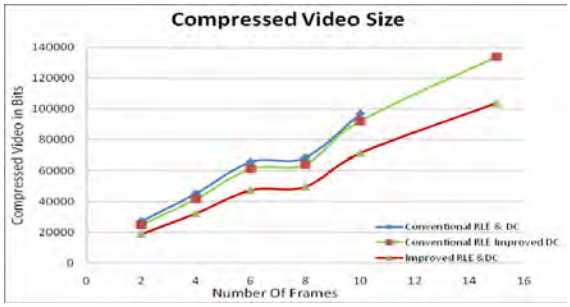


Figure 5. Comparison of compressed Video Size

Finally, the reduction in the size of the compressed video by using the proposed algorithm can be observed from Figure 5. The comparison based on PSNR is shown in Figure 6. It is very much evident that in spite of using different techniques to increase the compression, there is no or very little change in the PSNR of the reconstructed video and it is maintained at around 48 dBs. This indicates that the reconstructed video is of very good quality.

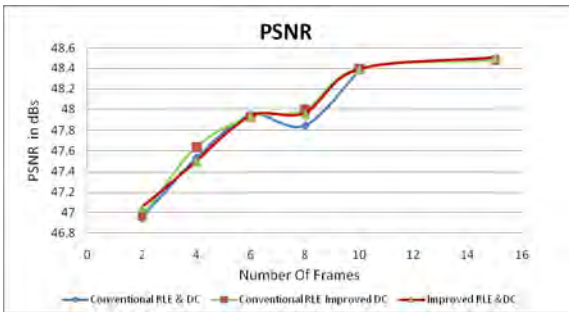


Figure 6. PSNR comparison

Further,the scale of quantization was increased from 1 to 5 for 15 frames of input video.The result of varying the quantization scale is depicted in the table 3(a)and(b).It is observed that by increasing the scale of quantization, the bit stream and the dictionary size of the compressed video reduces considerably while maintaining a good PSNR.

Table 3.(a) PSNR vs Quantization Scale

.

PSNR	
Quantization Scale=1	48.5058
Quantization Scale=2	47.0732
Quantization Scale=3	46.0427
Quantization Scale=4	44.0471
Quantization Scale=5	42.1601

Table 3(b) Bitstream vs Quantization Scale

Bit Stream Size	
Quantization Scale=1	98339
Quantization Scale=2	89466
Quantization Scale=3	83506
Quantization Scale=4	77998
Quantization Scale=5	72108

Since the PSNR is in the acceptable range for even the quantization scale of 5, depending on the required picture quality one can chose the scale and the compression ratio. In the next section, conclusive remarks are given.

6. Conclusions

In this paper, use of Accordion technique for video compression is presented. This technique consists of exploiting the high amount of temporal redundancies present in videos by converting them to spatial redundancy and using 2D DCT. Also, the conventional approaches related to Zigzag processing and Run Length Encoding are re-designed to get a further optimized Huffman dictionary. All the algorithms are developed in MATLAB environment. On comparing the conventional techniques and the proposed algorithm, a significant reduction of 60% in size of Huffman dictionary and 25% reduction in code word length are found by processing the DC components in this unique way. This results in a significant reduction in the size of compressed video while maintaining the PSNR at the same level (around 48db). The subjective quality of video is observed by varying the quantization scale. The quantization scale is varied from 1 to 5, and it has been observed that even with a scale of 5 the reconstructed video is of good visual quality. This technique can be effectively used for slow moving objects such as video conferencing, surveillance etc. However, the rest of optimization techniques will yield a significant additional compression without losing the video quality measured in PSNR.

References

[1] Thomas Wiegand and Gary J. Sullivan," —The H.264/AVC Video Coding Standards"; *IEEE Signal Processing Magazine*, March 2007.

[2] Gary j. Sullivan, Senior member, IEEE and Thomas Wiegand, "Video Compression—From Concepts to the H.264/AVC Standard", 0018-9219 © 2005 IEEE.

[3] Ashraf A. Al-Najdawi, Roy S. Kalawsky," Multiple objective optimization framework for Video compression and transmission,"978-1-4244-18763/08©2008 IEEE.

[4] Xiaoling Qiu, Haiping Liu, Dipak Ghosal and Biswanath Mukherjee, John Benko, Wei Li and Rashmi Bajaj, "Adaptive Video Compression Rate Optimization in Wireless Access Networks",978-1-4244-4487-8/09 ©2009 IEEE.

[5] Antonio Silva and Antonio Navarro," Ultra Low Bitrate Video Coding ",0-7803-8920-4/05©2005 IEEE.

[6] G.J. Sullivan and R.L. Baker,"Motion compensation for video compression using control grid interpolation", Proc. IEEE Int. Conf. Acoustics Speech, and Signal Processing (ICASSP), 1991, pp 2713–2716.

[7] G.J.Sullivan,"Multi-hypothesismotion compensation for low bit rate video coding", Proc. IEEE Int. Conf. Acoustics, Speech, and Signal Processing, 1993, pp 437- 440.

[8] Tarek Ouni, Walid Ayedi, Mohamed Abid, "New low complexity DCT based video compression Method",978-1-4244-2937-0/09 © 2009 IEEE.

Biographies

Ms. Poorva Waingankar is currently working as Associate Professor in Electronics Engineering Department at Thakur College of Engg. &Technology, Mumbai, India and pursuing Ph.D. in Electronics Faculty at Swami Ramanand Tirth Marathwada University, Nanded, India.

Dr.Sangeeta Joshi is currently Professor and Technical Advisor at Vidyalankar Institute of Technology, Mumbai, India. She has published number of research papers in reputed journals and received many awards. Her major fields of interest are, Embedded Systems and PC Based Instrumentation, VLSI Design, Reconfigurable Computing, Sensor Network, Simulation and Modeling of Nanodevices.

www.icgst.com

A Precise Technique for Hand Gesture Recognition

Ayush Purohit[1], Shardul Singh Chauhan[2]
Software Engineer, Larsen and Toubro Infotech, Mumbai, India[1]
Assistant Professor, Department of CSE, Northern India Engineering College, New Delhi, India[2]
ayushpurohit@outlook.com, shardulchauhan007@gmail.com

Abstract
Vision based methodologies provides a more natural and proficient result when contrasted with traditional strategies which have been utilized for hand gesture recognition. In this paper, we proposed a video based hand gesture recognition. Our approach commences by acquiring the video frame from a source and converting it into 2D binary frame using YCbCr color space. We implemented opening and closing operations to filter the noise from the frame. In order to track and segment the hand gesture we used Kalman filter and convex hull along with convexity defects for detecting hand regions from the frame. Our framework can perceive six kinds of hand gestures at present time.

Keywords: *Computer Vision, Convex Hull, Convexity Defect, Kalman Filter.*

Nomenclature

SVM	Support Vector Machine
CNN	Convolutional Neural Network
SAD	Sum of Absolute Differences
HOG	Histogram of oriented Gradients
PCA	Principal Component Analysis
LDA	Linear Discriminant Analysis

1. Introduction
Gesture recognition is a process of deciphering and comprehending the human gestures by implementing various algorithms. Gesture Recognition has been an area where a colossal measure of examination has been done which has numerous applications. An assortment of methodologies has been proposed for the procedure of gesture recognition. Data glove based methodology makes the utilization of sensor gadgets for digitalization of both hand and additionally finger movements into multi parametric information. Movement construct hand division approaches depend with respect to the supposition that the elements vital for gestures will be connected with gestures. Vision-based methodologies share the issue identified with the caprices of low-level division. Most of the image processing techniques are in light of two fundamental techniques: machine learning and rules.

A vision based hand gesture recognition system is proposed in [1] which uses scale space highlight discovery. In this work the first step is to make use of a specific hand gesture in order to detect the hands followed by tracking. The segmentation of hands is done using color cues and motion. Finally a scale space feature detection technique is used for integration in recognition of gestures. Jesus et al, in [2] examines depth based hand gesture recognition. The point has been to highlight the gesture classification strategies and additionally hand restriction techniques. Here a detailed study of 37 papers have been made for comparing various depth based gesture recognition systems on the basis of various aspects like hand localization, the effects of low cost Kinect, OpenNI software libraries and so on. A video based hand gesture recognition method has been implemented in [3]. The work focuses on recognition of hand gestures on a video stream. The proposed system focuses on two procedures namely the hand gesture detection and hand gesture recognition. The hand detection begins by locating the hands in the video frames with the help of blue rectangles by implementing Viola Jones technique. The hand gesture recognition begins with the Hu invariant moments feature vectors which are extracted from the detection of hand gestures and then trained and classified using SVM.

Another methodology is proposed in [4] utilizes modified census transform to highlight extraction process for gesture recognition. The claim to fame of the transform is that it is enlightenment invariant. Finally, a direct classifier is used for recognizing hand gestures. A video based hand gesture recognition technique is suggested in [5]. Initially a user hand gesture video is captured and stored in the hard disk. The videos captured are read by the system one by one and converted in the form of binary images. Then a 3D Euclidian space is created of the binary values obtained. For the training a feed forward neural network training method and for classification back propagation neural network is used. In [6], gesture recognition method is proposed which uses feed forward neural networks alongside back propagation for classifying the extracted features. The work compares various hand gesture recognition techniques by making the use of MATLAB. The use and implementation of skin detection and edge detection algorithms are also studied. Reference in [7], concentrates on the utilization of CNN for hand gesture recognition by making use of images captured by camera. To make the system robust, calibration of hand position, orientation and skin model are applied for obtaining the training as well as testing data for CNN. The Gaussian mixture model algorithm is used for training of the skin model. The calibrated images so obtained are used for the purpose of training the CNN.

Xianghua Li proposed thinning method which involves SAD to compute matching regions [8]. A depth map is implemented in the portion of hand detection that makes the use of sum of absolute differences technique for detection of the object located in foreground. The frame is converted into YCbCr space and then convex hull is computed to extract region of interest. The background image in the obtained region of interest is removed so that the foreground image can

be received that is hand image. A blob labeling method algorithm is used for obtaining the clear hand image. The feature point extracted using thinning algorithm is used to recognize hand gesture. Similar approach is used by Amiraj in [9], uses convex hull and convexity defects to count the number of fingers in video. The primary step is to capture the video and use it as an input for the system. The video is converted into frames and thresholding is applied to separate the hands from the background. Contours are used to find out the location of hands in the video frames. The algorithms like convex hull and convexity defects are implemented for detection and extraction of hands from the input. Then by making the use of various rules the hand gestures are classified. In [10], proposed automated method to recognize hand gestures in varying backgrounds. Skin color detection method has been used to figure out the hand region from the complex background. A series of morphological operations are implemented to extract the contour which is used to recognize finger tips. The angle of the fingertips is used for marking the fingertips. The technique shows the accuracy of the system with low computation cost. Yafei used HOG transform to extract hand features which are then reduced to 9D subspace using PCA-LDA [11]. The hand regions are finding out by combining an adaptive skin color detection algorithm along with the motion detection. The distance between the features of projections and each class of gesture is calculated. The extracted features are then classified using nearest neighbor to identify the gesture. The use of hands instead of mouse as an input appears to be an instinctive choice for man machine interaction.

In this paper, we used convex hull and convexity defects to describe the hand gestures. The hands are firstly detected using skin color and various morphological operations are used to extract the features using convex hull properties. For the purpose of tracking, Kalman filter is used to track the location of hands in the video frames. The classification is done on the basis of the specified rule set. Finally the results of the proposed technique have been tabulated which indicates the precision of the system.

2. Hand Detection

In order to locate the hand gesture in a video frame efficiently, skin color detection and region of interest are computed.

Skin Color Detection

Skin color detection is a procedure of identifying the region of interests within the spectrum of skin colored pixels in an image or a video frame. This methodology is utilized in various approaches which incorporate distinguishing a face, object, hand, etc. in diversified expanses.

Due to vacillating background conditions & luminance components, we erected our skin color model in YCbCr color space in order to approximate the chromaticity of skin. This computation involves conversion of RGB to YCbCr color space and eliminating the luminance component to compose the skin color more robust to illumination. The histogram of the resulting 2D color vector has produced the region of interest which shows a strong peak at the skin color. This conversion step is explained using a diagram as represented in figure 1.

The YCbCr conversion of a given pixel from RGB can be deduced by the following matrix I:

$$\begin{bmatrix} Y \\ Cr \\ Cb \end{bmatrix} = [\mathrm{R\ G\ B}] \begin{bmatrix} 0.299 & -0.1689 & 0.4998 \\ 0.587 & -0.3317 & 0.4185 \\ 0.114 & -0.5006 & -0.0813 \end{bmatrix} \quad \text{(I)}$$

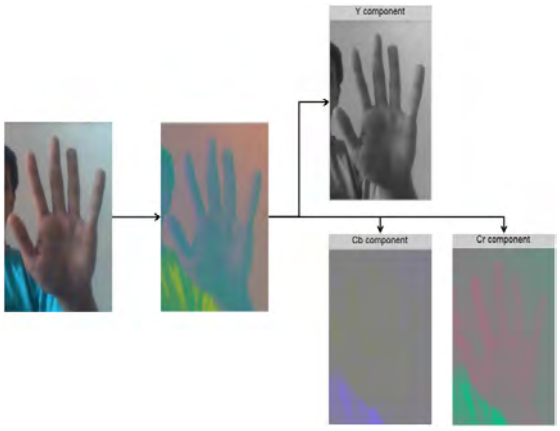


Figure 1: Step1: Conversion of RGB to YCbCr color space. Step 2: Separating Y, Cb and Cr components from YCbCr frame.

3. Motion Detection

To the resultant 2D gray scale color vector, morphological transformations are performed. We initiated the process by thresholding the grayscale framework. This method reorganizes a grayscale image to a bi-level image and extracts the pixels representing the hand or an object. A median filter with a kernel 15 x 15 is used to filter the noise from the resulted frame. A combination of morphological operations which consist of binary opening and closing, are applied over the image to suppress the remaining noise using a square kernel.

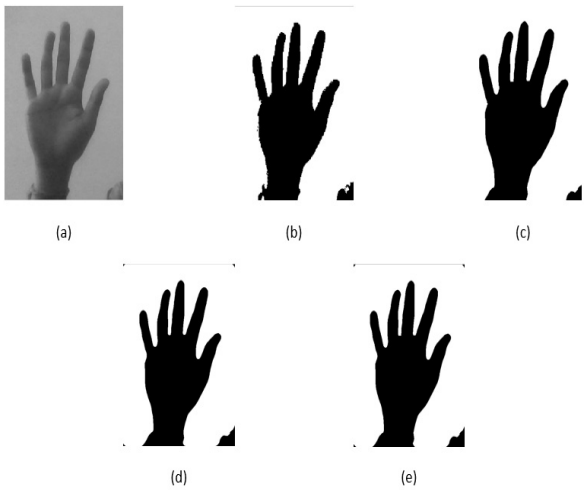


Figure 2: a) Grayscale Image. b) Threshold Operation. c) Median Filter Operation. d) Opening Operation. e) Closing Operation.

The opening of image I by kernel H can be computed as:

$$(I o \mathrm{H}) = (I \oplus H) \Theta H \quad (1)$$

To the resulting frame, we performed thresholding operation to acquire an optimal frame for computing hand gesture features. A series of morphological operations implemented over the video frame is shown in figure 2.

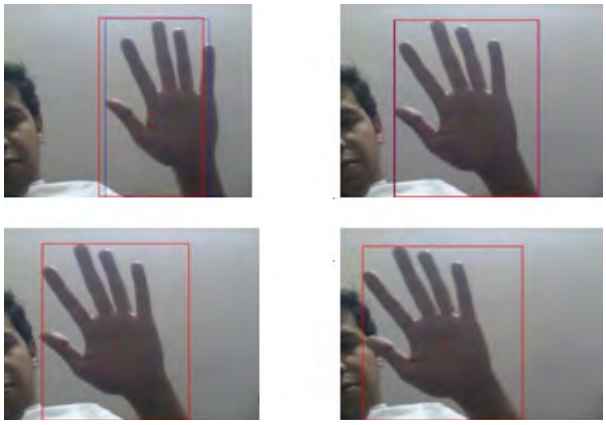


Figure 3: Sequence of frames extracted from video using Kalman Filter.

Hand Tracking

To track hand gestures in real time, we implemented an optimal estimation framework catered by Kalman filter which is extensively adopted for tracking objects because of its small computational requirements, elegant recursive properties, uncertainty analysis and prognosis of subsequent frames [12] [13]. In this paper, Kalman filter is employed to predict the location of hand gesture in a frame. The Kalman filter follows a two-step procedure for hand tracking, that are control and measurement update. The control update can be used for estimation of the state with the previous state and vector, while the measurement update is used for correcting the sensor information based upon the state. To finally predict the position of hand in the frame, we blend the Gaussian results produced from prediction and measurement to obtain the position of the hand as shown in figure 3.

4. Projection into Palm Plane

To project into palm plane, contours are configured around the black bead of the hand developed after segmenting the frame. It is possible that system might detect multiple contours which are produced due to noise in the background. An assumption is made that contours produced by the noise are smaller in size compared to contour of the hand. Therefore, we scrutinized the biggest contour in the frame which is used for further processing. This method thus removes the possibility of considering any contour formed due to noise.

Convexity Detection

The final approach of our system is to detect convexity points from the extracted contour. This methodology endeavors to detect convex hull and convexity points from the contour. The convex hull illustrates the extrinsic contour of the hand such that all the contour specks are within the convex hull.

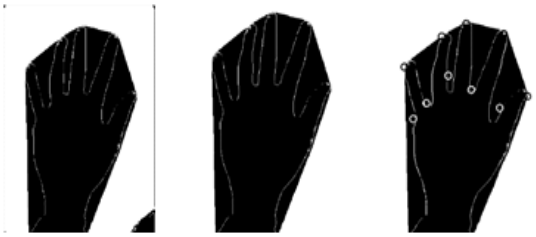
To extract the convex hull, we approximated the hand contour with a minimum parameter polygon resulting in dwindling of undesirable convexity specks. We used Douglas-Peucker algorithm for smoothing the boundary which recursively joins first and last vertices of the polygonal line segment by finding the vertex furthest from it.

To estimate convex hull points of the approximation polygon, we implemented a simple and intuitive Sklansky's algorithm. This graph based algorithm is based on stack, which in the extreme includes the vertices of the convex hull. It considers three vertices: top stack vertex, new vertex, second to top of the stack vertex. The top stack vertex is rebuffed if trio forms a right turn.

Convexity defects are computed by measuring distance between the farthest point and convex hull. The resulting frame is filtered by rejecting the convex points which are not present near finger tips. This is done by computing the centroid of enclosed polygon. If any convex point whose height is less, then height of the center of the palm was filtered out.

Hand Gesture Recognition

This application is developed to identify the number of fingers operating in a hand gesture. To classify the number of fingers distinguishable in the frame, we used feature extracted from frames and counted the number of convex and convexity defect points. Figure 4 indicates the use of convex hull and convexity defects to find out the hand points that are needed for recognizing hand gestures.



(a) (b) (c)

Figure 4: (a) Convex Hull of the frame. (b) Extracted Contour. (c) Convex and Convexity Defect Points.

Finger Counting

Using polylines drawn around the hand, we computed the approximate centroid of the hand. For any of the parameter to correctly satisfy the prerequisite, the 'x' number of convex hull points should lay outside a threshold range from the centroid of the hand. In order to recognize the number of fingers, one of the following parameters should be satisfied as shown in Table 1:

No. of fingers	Convex Hull Points (*x*)	Convexity Defect Points (*y*)
0	Exactly 0	Exactly 0
1	Exactly 1	Exactly 0
2	Exactly 2	At least 1
3	Exactly 3	At least 1
4	Exactly 4	At least 1
5	Exactly 5	At least 1

Table 1: Condition for recognizing finger counts

5. Experimental Results

In this model, there are certain constraints which need to be satisfied for recognizing the hand gesture and count the number of fingers. The system also maintains the tracks of the hand gesture which uses Kalman filter. In figure (5) and (6), shows the current working model which can trail the hand and recognizes limited number of finger counts. In order to find out the classification rate of the system a set of 20 videos are used for each hand gesture. The aim was to ensure that the arrangement of videos contain enough data with a specific end goal to depict a specific hand gesture.

The set involves videos which delineates a solitary hand performing gestures where hand ought to possess the significant locale. Table 2 indicates the classification results of the system.

		Result of Classification						Unrecognized	Error Percentage
Inputs	Class of Gestures	0	1	2	3	4	5		
	0	19	0	0	0	0	0	1	5
	1	0	20	0	0	0	0	0	0
	2	0	0	20	0	0	0	0	0
	3	0	0	0	20	0	0	0	0
	4	0	0	0	0	20	0	0	0
	5	0	0	0	0	0	20	0	0

Table 2: Classification Results

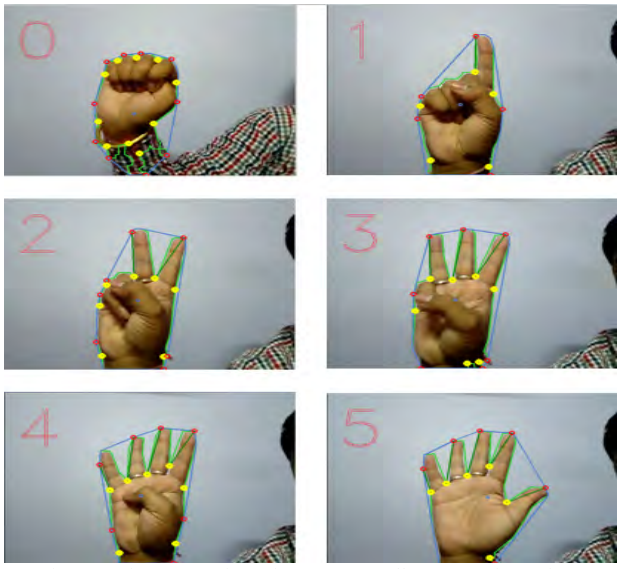


Figure 5: Finger Counting using Convex Hull & Convexity Defects.

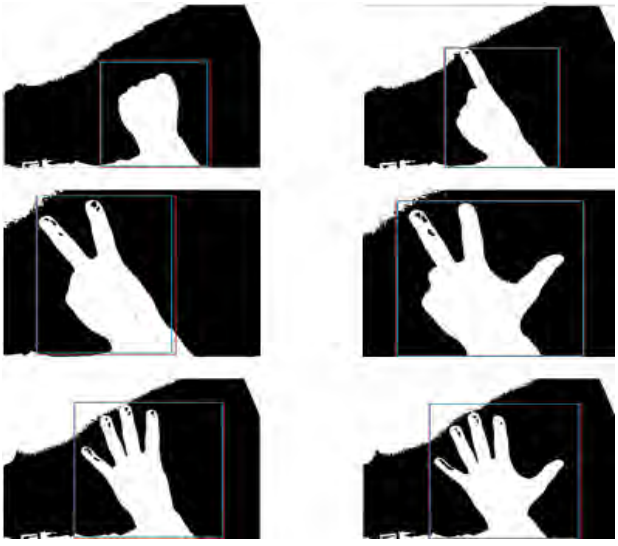


Figure 6: Hand Tracking in Binary Video Using Kalman Filter.

6. Conclusion and Future Work

In this paper, we presented a vision-based hand gesture recognition system which operates on real time videos on an average PC using low cost cameras. The proposed method is currently used to count limited number of fingers with a high classification rate under various constraints. The future work involves recognizing multiple hands in a given frame, a rotation and orientation independent gesture recognition and a more efficient and flexible man-machine interaction which can be used in real life applications.

References

[1] Yikai Fang, Kongqiao Wang, Jian Cheng, Hanqing Lu, A Real-Time Hand Gesture Recognition Method, *Proc. IEEE International Conference on Multimedia and Expo*, 2007, 995-998.

[2] Jesus Suarez, Robin R. Murphy, Hand Gesture Recognition with Depth Images: A Review, *Proc. IEEE Ro-Man: The 21st IEEE International Symposium on Robot and Human Interactive Communication*, 2012, 411-417.

[3] Liu Yun, Zhang Peng, An Automatic Hand Gesture Recognition System Based on Viola-Jones Method and SVM's, *2nd IEEE International Workshop on Computer Science and Engineering*, 2009, 72-76.

[4] Agnes Just, Yann Rodriguez, Sebastien Marcel, Hand Posture Classification and Recognition using the Modified Census Transform, *Proceedings of the 7th IEEE International Conference on Automatic Face and Gesture Recognition*, 2006, 351-356.

[5] Murthy, G.R.S, Jadon, R.S, Hand gesture recognition using neural networks, *Proc. IEEE 2nd International Advance Computing Conference*, 2010, 134-138.

[6] Yewale, S.K, Bharne, P.K, Hand gesture recognition using different algorithms based on artificial neural network, *IEEE International Conference on Emerging Trends in Networks and Computer Communications*, 2011, 287-292.

[7] Hsien-I Lin, Ming-Hsiang Hsu, Wei-Kai Chen, Human hand gesture recognition using a convolution neural network, *IEEE International Conference on Automation Science and Engineering*, 2014, 1038-1043.

[8] Xianghua Li, Jun-ho An, Jin-hong Min and Kwang-Seok Hong, Hand gesture recognition by stereo camera using the thinning method, *IEEE International Conference on Multimedia Technology (ICMT)*, 2011, 3077-3080.

[9] Amiraj Dhawan, Vipul Honrao, *Implementation of Hand Detection based Techniques for Human Computer Interaction*, *International Journal of Computer Applications, 18(6)*, 2013, 0975 – 8887.

[10] Xiaotang Wen, Yanyu Niu, A method for hand gesture recognition based on morphology and Fingertip-Angle, *IEEE 2nd International Conference on Computer and Automation Engineering (ICCAE)*, 2010, 688-691.

[11] Yafei Zhao, Weidong Wang, Yuehai Wang, A real time hand gesture recognition method, *IEEE International Conference on Multimedia and Expo*, 2007, 995-998.

[12] C. K. Chui, G. Chen, *Kalman Filtering with real-time applications* (Berlin, Springer-Verlag).

[13] Shiuh-Ku Weng, Chung-Ming Kuo, Shu-Kang Tu, Video object tracking using adaptive Kalman filter, Journal of Visual Communication and Image Representation, volume 17, issue 6, Dec 2006.

Biographies

Mr. Ayush Purohit is currently working as a Software Engineer in Larsen and Toubro Infotech, Mumbai. He completed his BTech in Computer Science and Engineering and MTech in Artificial Intelligence and Artificial Neural Networks from University of Petroleum and Energy Studies, Dehradun. His area of interest includes Computer Vision, Data Visualization and Business Process Automation.

Mr. Shardul Singh Chauhan is currently working as an Assistant Professor in Northern India Engineering College, New Delhi. He completed his BTech in Computer Science and Engineering and MTech in Artificial Intelligence and Artificial Neural Networks from University of Petroleum and Energy Studies, Dehradun. His area of interest includes Digital Image Processing, Artificial Neural Networks, Expert Systems and Pattern Recognition.

www.icgst.com

Multidimensional Approach for Securing Images on Cloud

D. Boopathy[1], M. Sundaresan[2]
[1 & 2]*Department of Information Technology, Bharathiar University, Coimbatore- 641046,Tamilnadu, India*
ndboopathy@gmail.com

Abstract

Encryption is one of the methodologies used to maintain and protect the data confidentiality. As per the user data type's requirements, users need to adopt and implement any one of the existing methods. But those encryption methods and standards may not be bound within the user data country regulations, when the users are from different geographical locations. Some of the existing methods are already compromised by hackers and also some of the government agencies are forcing their country based service providers to provide the encrypted information in the name to maintain the country's security. It is very difficult to manage the threats with one method. The proposed method tried its maximum level to reduce the threats by using different points of view. In this proposed method images and the block-based encryption method have been used to protect the normal and sensitive image from the unauthorized access. The proposed method is tested on all proposed encryption types using greyscale in two scenarios. They are Different Images One Type (DIOT) and Single Image All Types (SIAT). The results of the proposed methods are evaluated using PSNR, MSE, Size of the Image and Histogram to verify the image's integrity.[1]

Keywords: *Image Encryption, Decryption, Image Security, Greyscale Images, Cloud Security.*

Nomenclature

SCDSPM	Secured Cloud Data Storage Prototype Model
E & DGM	Encryption & Decryption Gateway Model
MDE & DPM	Multi-Dimensional Encryption & Decryption Model
PRA	Pixel Rearrange Algorithm
PRRA	Pixel Reverse Rearrange Algorithm
PSA	Pixel Shuffling Algorithm
PRSA	Pixel Reverse Shuffling Algorithm
DIOT	Different Images One Type
SIAT	Single Image All Types

[1]This study has been implemented and Tested on Java platform at Department of Information Technology, Bharathiar University, Coimbatore, Tamilnadu, India.

1. Introduction

"One picture is worth a thousand words", is a popular English saying which has been used since 1918, in a newspaper advertisement for the San Antonio Light [1]. The image conveys the complete information to the viewers without any loss of any piece of information. Sensitive images must be safeguarded from the general viewers and unauthorized viewers in order to protect the confidential nature of its contents. For this purpose, different encryption standards are applied on the sensitive and non-sensitive images to maintain the image confidentiality and to prevent that image from being mishandled by the unauthorized and unidentified users. While coming to the specific objective, the existing encryption standards in use are not reliable due to their limitations, data processing technique and algorithm working architecture methods. Once the images are stored online, then the owner of the images automatically loses his rights on those images. The online service providers are altering their policies in data handling and even reformatted the policy related data from time to time without any users' interactions. That service provider's server may be geographically positioned in some other vicinity and in that place only the encryption and decryption will take place. Once the user encrypts the data by using a specific service provider, then the user needs to decrypt that data by using that same service provider only but it may be done from anywhere because they are online. If the user is using the offline encryption tools, then the user needs to depend on that device for the encryption and decryption, but the user must always keep the device with him to perform either encryption or decryption whenever necessary. Taking all these things into consideration, the Secured Cloud Data Storage Prototype Model is designed and the Multi-Dimensional Encryption and Decryption Method is one of the modules in that. Section 2 reviews the related works concerning the encryption techniques to maintain the security of the data storage. Section 3 deliberates on the different working methodology, procedure, Pseudo code and Testing file details of MDE & DPM Algorithm. Section 4 delineates the implementation, and Section 5 explains the experimental results and in addition the features of the proposed method are also discussed here. Section 6 presents the conclusion derived from the findings, the

advantages of the proposed algorithm and finally its related future enhancements.

2. Literature Review

New image encryption design which utilizes one of the three dynamic chaotic [2] systems to shuffle the location of the image pixels and uses another one of the same three chaotic maps to mystify the association between the cipher image and the plain-image, thereby considerably increasing the resistance to attacks. To overcome this, Sakthidasan et al proposed the algorithm with the advantage of bigger key space, lesser iteration times and high security analysis such as key space analysis, statistical analysis and sensitivity analysis [3].Navitha et al proposed a very new and combined approach for DCT based image compression, pixel shuffling based encryption, decryption and steganography for real-time applications [4]. Quist et al suggested the sets out method to contribute to the general body of knowledge in the area of cryptography application by developing a cipher algorithm for image encryption of m*n size by shuffling the RGB pixel values. The algorithm ultimately makes it possible for encryption and decryption of the images based on the RGB pixel [5]. Junqin et al introduced a permutation-substitution image encryption scheme based on generalized Arnold map. Only one round of permutation and one round of substitution are performed to get the desirable results. The generalized chaotic Arnold maps are applied to generate the pseudo-random sequences for the permutation and substitution [6]. Lohit et al explored the implementation of AES in MATLAB on plaintext encryption and cipher text decryption. These results are superior to the similar software implementations of AES [7].

3. Methodology

The existing methods, updated algorithms are using different concepts and implementations of these are enough to handle the data encryption process in offline mode but not in online mode. In online mode, i.e.in cloud [8], the existing methods require more time and utilize more resources to perform the encryption and decryption process. The geographically distributed data processing servers will raise the security breach issues and data trans-border related issues. So, the data need to be encrypted before the data are transferred from the user end to the server end. The proposed method considered all of these measures and provides the prototype model with different modules to overcome the data related storage, retrieval and encryption issues.

SCDSPM

The Secured Cloud Data Storage Prototype Model [9, 10] contains four sub-modules; they are:

- Authentication Authorization Resolving Module [11, 12]
- Data Type Identification and Extension Validation Module [13]
- Encryption and Decryption Gateway Module [14 - 16]
- Automatic Cloud Data Backup Module [17]

This paper explains the third module of SCDSPM i.e. E & DGM. This E & DGM is redefined with some modification and named in this paper as Multi-Dimensional Encryption and Decryption Module (MDE & DPM). Figure 1 shows the proposed Multi-Dimensional Encryption and Decryption Module (MDE & DPM).

Multi-Dimensional Encryption and Decryption Module framework

Using the new type of encryption method will avoid the user's data from superfluous risks. Each and every encryption and decryption logics must be uniquely different from other methods. In that way, the proposed encryption algorithm is using new logic and it will help to avoid the unconstitutional access, illicit usage and unlawful surveillance of the user's data by unauthorized persons.

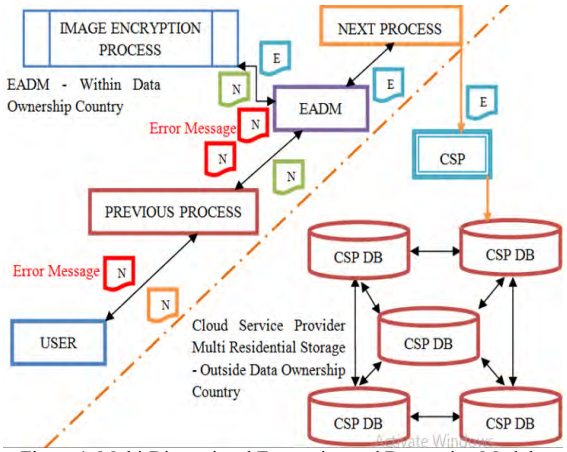


Figure 1. Multi-Dimensional Encryption and Decryption Module

The proposed Multi-Dimensional Encryption and Decryption Module presently concentrated on image format files only. This paper explains the proposed Multi-Dimensional Encryption and Decryption Module with tested standard and non-standard images and its related experimental results. It uses 512 x 512 pixel [18] images for testing purposes. Multi-Dimensional Encryption and Decryption Module contains four different algorithms to encrypt and decrypt the image. The four algorithms are: 1. Pixel Rearrange Algorithm, 2. Pixel Shuffling Algorithm, 3. Pixel Reverse Rearrange Algorithm, 4. Pixel Reverse Shuffling Algorithm.

The above mentioned algorithms are tested with different test case images which include standard images and non-standard images. Figures 2(a) and 2(b) shows the MDE & DPM's encryption and decryption method.

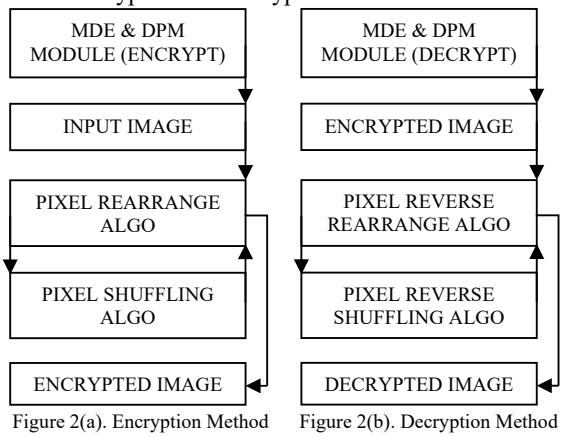


Figure 2(a). Encryption Method

Figure 2(b). Decryption Method

Pixel Rearrange Algorithm (PRA)

In Pixel Rearrange Algorithm the image pixels are rearranged into different positions using the 4 X 4 matrix concept. The pixel values of the images are relocated to other positions from their original positions. Once the image pixels are relocated to another position, then they automatically reflect in the original structural content of the image. Pixel Rearrange Algorithm is holding 4096 possible ways to rearrange the image pixels into a new position within the selected 4 X 4 matrix method. The result obtained from PRA is incorporated into the PSA.

1	2	3	4	R_n
5	6	7	8	R_n
9	10	11	12	R_n
13	14	15	16	R_n
C_n	C_n	C_n	C_n	C_n / R_n

1	9	10	8	R_n
16	2	7	11	R_n
14	6	3	12	R_n
5	15	13	4	R_n
C_n	C_n	C_n	C_n	C_n / R_n

Figure 3. Before applying PRA and after applying PRA

Figure3 shows the image pixel location before applying Pixel Rearrange Algorithm (PRA) and also shows the image pixel location after applying Pixel Rearrange Algorithm (PRA).

Pixel Reverse Rearrange Algorithm (PRRA)

The PRRA algorithm is used to reverse the Pixel Rearrange Algorithm's (PRA) relocated pixel values into their original position i.e. original location. The reversing method will use the rearrange method information from the decryption key.

1	9	10	8	R_n
16	2	7	11	R_n
14	6	3	12	R_n
5	15	13	4	R_n
C_n	C_n	C_n	C_n	C_n / R_n

1	2	3	4	R_n
5	6	7	8	R_n
9	10	11	12	R_n
13	14	15	16	R_n
C_n	C_n	C_n	C_n	C_n / R_n

Figure 4. Before applying PRRA and after applying PRRA

Figure 4 shows the pixel location before applying Pixel Reverse Rearrange Algorithm (PRRA) and also shows the pixel location after applying Pixel Reverse Rearrange Algorithm (PRRA).

Pixel Shuffling Algorithm (PSA)

The Pixel Shuffling Algorithm (PSA) is used to shuffle the pixel values within the matrix value. This research work holds sixteen different types of pixel values shuffling methods. Within those different methods, one of the methods will be automatically (i.e. randomly) selected and applied by the Pixel Shuffling Algorithm (PSA); then the selected method results will be stored with the decryption key. In each and every pixel shuffling method, one of the value locations will be fixed as a constant to identify which shuffling method is used to shuffle the pixel values. Here the decryption key will be automatically generated by the PSA algorithm with PSA related information and that information will be used at the time of decryption.

1	2	3	4	R_n
5	6	7	8	R_n
9	10	11	12	R_n
13	14	15	16	R_n
C_n	C_n	C_n	C_n	C_n / R_n

9	1	10	8	R_n
16	2	11	7	R_n
15	6	12	3	R_n
5	14	13	4	R_n
C_n	C_n	C_n	C_n	C_n / R_n

Figure 5. Before applying PSA and after applying PSA

Figure5 shows the pixel location before applying Pixel Shuffling Algorithm (PSA) and also shows the pixel location after applying Pixel Shuffling Algorithm (PSA). In the selected pixel shuffling method, the pixel value 14 is fixed as a constant value to identify the shuffling method.

Pixel Reverse Shuffling Algorithm (PRSA)

The decryption key holds the used Pixel shuffling algorithm's information. By using that information only the pixel reverse shuffling algorithm will work. Once the Pixel Reverse Shuffling Algorithm (PRSA) gets the information from the decryption key, then it will apply that correlated reverse shuffling method on that shuffled image pixel values. Once the pixel values are reversed, then it needs to be processed with the Pixel Reverse Rearrange Algorithm (PRRA). Then only the original structured content of the image will be constructed.

9	1	10	8	R_n
16	2	11	7	R_n
15	6	12	3	R_n
5	14	13	4	R_n
C_n	C_n	C_n	C_n	C_n / R_n

1	2	3	4	R_n
5	6	7	8	R_n
9	10	11	12	R_n
13	14	15	16	R_n
C_n	C_n	C_n	C_n	C_n / R_n

Figure 6. Before applying PRSA and after applying PRSA

Figure 6 shows the pixel location before applying Pixel Reverse Shuffling Algorithm (PRSA) and also shows the pixel location after applying Pixel Reverse Shuffling Algorithm (PRSA). By using the pixel value 14, which is fixed as constant value, is used to identify the shuffling method.

The Pixel Rearrange Algorithm (PRA) and Pixel Shuffling Algorithm (PSA) are used to encrypt the image. The Pixel Reverse Rearrange Algorithm (PRRA) and Pixel Reverse Shuffling Algorithm (PRSA) are used to decrypt the image.

Pseudo code for MDE&DMF

Encryption pseudo code:

Get the image from the user
Store that image into an **Object**
Read the Object Pixel Values
Store that Object Pixel Values into a Red, Green and Blue band color Text File
Get the Pixel Values from that Text Files
Store that Pixel Values of Text Files as three **Objects**
Apply the **PRA** on all the **Objects**
Apply the **PSA** on all the **Objects**
Apply the **PRA** on all the **Objects**
Prepare the **Decryption Key** with used algorithm method information
Convert all the Pixel Values Text Files and apply respective color band and merge all that color band files into an **Image File**
Store that **Image File** into selected storage in selected format
Store that **Image Decryption Key** into the selected storage in desired format

Decryption pseudo code:

Get the image from the user
Store that image into an **Object**

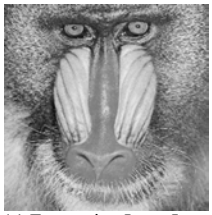
Get the Decryption Key to apply and decrypt the image
If the key got authenticated **Then**
Forward the process to next step
Else
Show an error message as key is invalid and **STOP** the process
Read the Object Pixel Values
Store that Object Pixel Values a Red, Green and Blue band color Text File
Get the Pixel Values from that Text Files
Store that Pixel Values of Text Files as three **Objects**
Apply the **PRRA** on all the **Objects**
Apply the **PRSA** on all the **Objects**
Apply the **PRRA** on all the **Objects**
Convert all the Pixel Values Text Files and apply respective color band and merge all that color band files into an **Image File**
Store that **Image File** into the selected storage in selected format

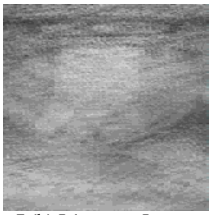
4. Implementation

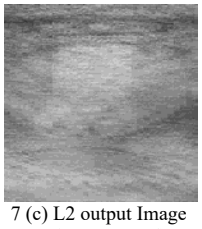
The proposed method has been implemented using the MATLAB simulation tool and Java 1.8 programming tool. The implementation is divided into two parts. The first part is to read and write the three band image pixel values into the text file and then it needs to read the three band image pixel values from the text file and to construct the image file. Five different standard images [19] and two non-standard images are taken for testing purpose. Each testing file contains 512 x 512 pixel image. The remaining details of the testing images are shown in Table 1.

Table 1. Testing Images

Image Sl. No.	Image Name	Standard / Normal Image	Image Size
1	Baboon	Standard Image / Tiff Format	258 KB
2	Cameraman	Standard Image / Tiff Format	256 KB
3	Lena	Standard Image / Tiff Format	260 KB
4	Pirate	Non-Standard Image / Tiff Format	257 KB
5	Room	Non-Standard Image / Tiff Format	258 KB
6	Peppers	Standard Image / Tiff Format	206 KB
7	House	Standard Image / Tiff Format	106 KB


7 (a) Encryption Input Image


7 (b) L1 output Image


7 (c) L2 output Image

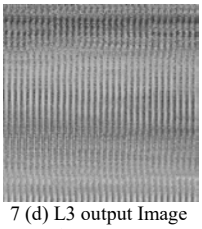
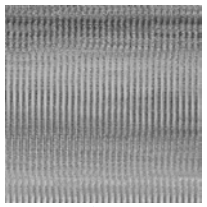
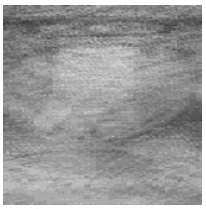
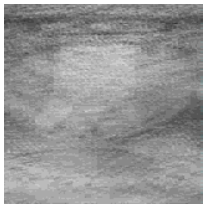

7 (d) L3 output Image

Figure 7 (a - d). Step by Step Encryption Process


8 (a) Decryption Input Image


8 (b) L1 output Image


8 (c) L2 output Image


8 (d) L3 output Image

Figure 8 (a - d). Step by Step Decryption Process

The step by step working formation on image of the proposed encryption and decryption algorithm is applied on the baboon standard "TIFF" [20] image file format [21] and the resultant images are shown above. Figure 7(a) is the input image, figure 7(b) is the first level output image, figure 7(c) is the second level output image, and figure 7(d) is the final level output image i.e., encrypted image. Similarly figure 8(a) is the encrypted image, figure 8(b) is the first level output image, figure 8(c) is the second level output image, and figure 8(d) is the final level output image i.e. decrypted image. There are sixteen different types of pixel shuffling algorithms are available in the proposed Pixel Shuffling algorithm. Among those pixel shuffling methods, for testing purpose all the algorithms are used in this paper.

5. Results and Discussion

There is no change found on the histogram of normal image, encrypted image and decrypted image. The histogram of Baboon image for normal image, encrypted image and decrypted image is shown below in figure 9 (a-c). The implementation is done in two ways, they are:

- Different Image One Type method (DIOT)
- Single Image All Types method (SIAT)

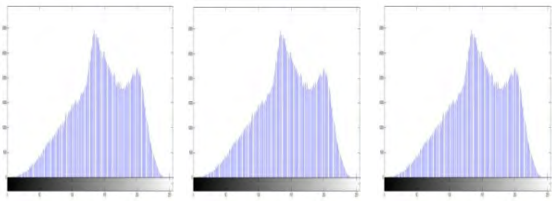

9 (a) 9 (b) 9 (c)

Figure 9 (a - c). Histogram of Normal, Encrypted and Decrypted Image

Figure 9(a) shows the normal input image histogram [22], Figure 9(b) shows the encrypted image histogram [22] and Figure 9(c) shows the decrypted image histogram [22]. Table 2 shows 7 different Input images and its related encrypted image and decrypted image in Different Images One Type method (DIOT). The different images are processed in one of the proposed encryption algorithms to verify the algorithm working style. In that, Type-16 encryption algorithm has been used to encrypt the test images. In the table 2, the 1a – 7a images are

normal input image, the 1b – 7b are encrypted image and 1c – 7c are decrypted image.

Table 2. DIOT related Normal, Encrypted and Decrypted Image

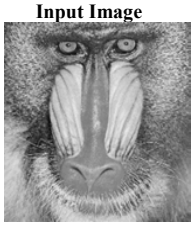
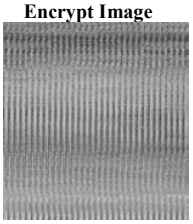
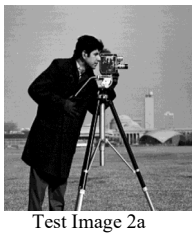
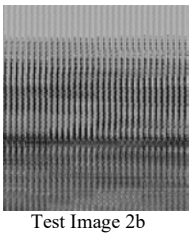
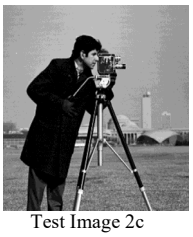
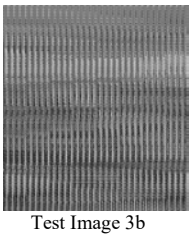
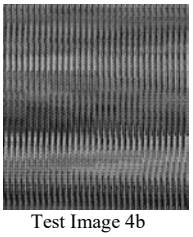
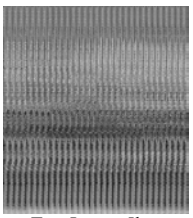
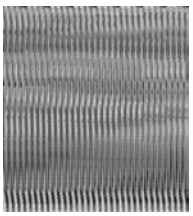
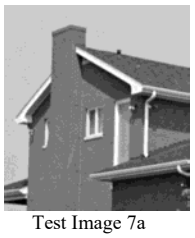
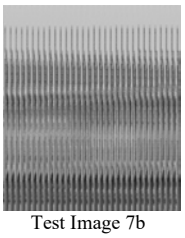
Input Image	Encrypt Image	Decrypt Image
 Test Image 1a	 Test Image 1b	 Test Image 1c
 Test Image 2a	 Test Image 2b	 Test Image 2c
 Test Image 3a	 Test Image 3b	 Test Image 3c
 Test Image 4a	 Test Image 4b	 Test Image 4c
 Test Image 5a	 Test Image 5b	 Test Image 5c
 Test Image 6a	 Test Image 6b	 Test Image 6c
 Test Image 7a	 Test Image 7b	 Test Image 7c

Table 4 shows the parameters used to verify the comparison between encryption image and decryption image with normal image in DIOT. The Size of the Image, isequal() Function [23], PSNR [24] and MSE [25] are taken as parameters and they are compared among input image, encrypted image and decrypted image.

Table 3 shows the input image and the proposed encryption algorithms (i.e. From Type-01 to Type-16) encrypted images in Single Image All Types method i.e. SIAT. All the encrypted images might look like the same, but the differences are present in the Type-01 to Type-16 encrypted image outputs. The parameters used to measure the differences between the encrypted images are shown in table 5. The Single image is processed in all the proposed encryption algorithms, i.e. Type-01 to Type-16 to verify that all the algorithm's encrypted images are different from one another or not.

Table 3. SIAT related Normal, Encrypted and Decrypted Image

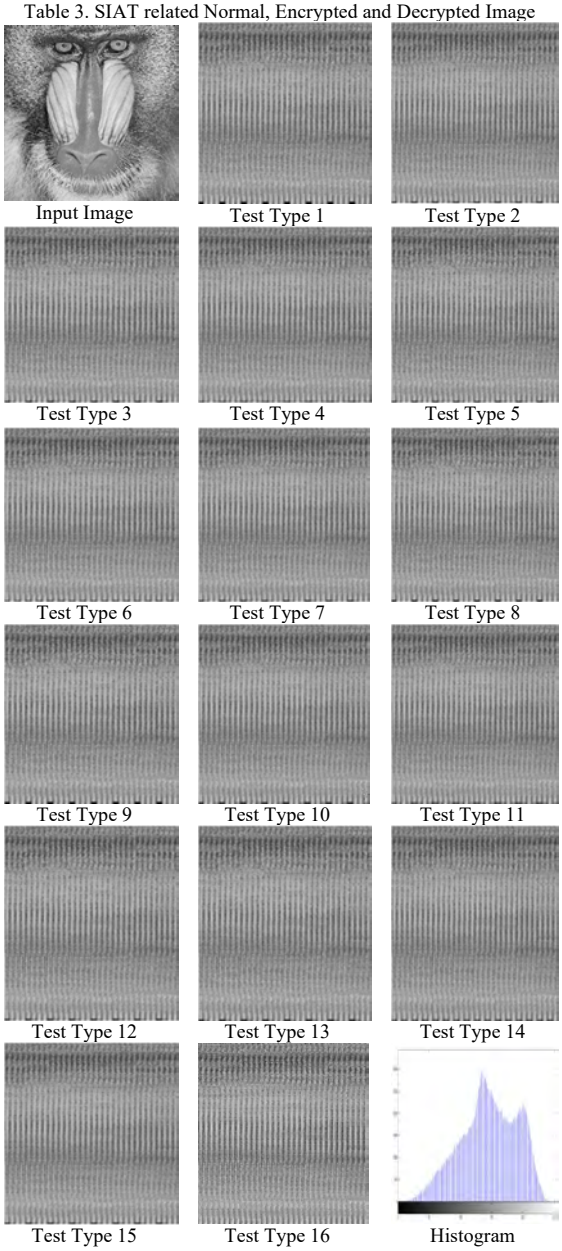


Table 4. Comparison of size of the image, isequal() Function, PSNR and MSE for Different Images One Type (DIOT)

Image Name	Details		Size of the Image	isequal ( ) Function	PSNR Value	MSE Rate
Test Image 1	Input Test Image Vs	Encryption Image	258 kB	0	33.8481328 dB	108.08
		Decryption Image	258 kB	1	Inf dB	0
Test Image 2	Input Test Image Vs	Encryption Image	256 kB	0	34.7574709 dB	87.66
		Decryption Image	256 kB	1	Inf dB	0
Test Image 3	Input Test Image Vs	Encryption Image	260 kB	0	33.7661045 dB	110.14
		Decryption Image	260 kB	1	Inf dB	0
Test Image 4	Input Test Image Vs	Encryption Image	257 kB	0	33.7901752 dB	109.53
		Decryption Image	257 kB	1	Inf dB	0
Test Image 5	Input Test Image Vs	Encryption Image	258 kB	0	34.0635209 dB	102.85
		Decryption Image	258 kB	1	Inf dB	0
Test Image 6	Input Test Image Vs	Encryption Image	206 kB	0	35.1889686 dB	79.37
		Decryption Image	206 kB	1	Inf dB	0
Test Image 7	Input Test Image Vs	Encryption Image	106 kB	0	33.7953091 dB	109.40
		Decryption Image	106 kB	1	Inf dB	0

Table 5. Comparison of size of the image, isequal() Function, PSNR and MSE for Single Image All Types (SIAT)

Algorithm Type	Details		Size of the Image	isequal ( ) Function	PSNR Value	MSE Rate
Type 1	Input Test Image Vs	Encryption Image	234 kB	0	27.7668073 dB	108.74
		Decryption Image	234 kB	1	Inf dB	0
Type 2	Input Test Image Vs	Encryption Image	234 kB	0	27.7640038 dB	108.81
		Decryption Image	234 kB	1	Inf dB	0
Type 3	Input Test Image Vs	Encryption Image	234 kB	0	27.7620891 dB	108.86
		Decryption Image	234 kB	1	Inf dB	0
Type 4	Input Test Image Vs	Encryption Image	234 kB	0	27.7621153 dB	108.86
		Decryption Image	234 kB	1	Inf dB	0
Type 5	Input Test Image Vs	Encryption Image	234 kB	0	27.7555824 dB	109.02
		Decryption Image	234 kB	1	Inf dB	0
Type 6	Input Test Image Vs	Encryption Image	234 kB	0	27.7684094 dB	108.70
		Decryption Image	234 kB	1	Inf dB	0
Type 7	Input Test Image Vs	Encryption Image	234 kB	0	27.7640273 dB	108.81
		Decryption Image	234 kB	1	Inf dB	0
Type 8	Input Test Image Vs	Encryption Image	234 kB	0	27.7596387 dB	108.92
		Decryption Image	234 kB	1	Inf dB	0
Type 9	Input Test Image Vs	Encryption Image	234 kB	0	27.7571000 dB	108.99
		Decryption Image	234 kB	1	Inf dB	0
Type 10	Input Test Image Vs	Encryption Image	234 kB	0	27.7681899 dB	108.71
		Decryption Image	234 kB	1	Inf dB	0
Type 11	Input Test Image Vs	Encryption Image	234 kB	0	27.7622032 dB	108.86
		Decryption Image	234 kB	1	Inf dB	0
Type 12	Input Test Image Vs	Encryption Image	234 kB	0	27.7632227 dB	108.83
		Decryption Image	234 kB	1	Inf dB	0
Type 13	Input Test Image Vs	Encryption Image	234 kB	0	27.7643787 dB	108.80
		Decryption Image	234 kB	1	Inf dB	0
Type 14	Input Test Image Vs	Encryption Image	234 kB	0	27.7658733 dB	108.77
		Decryption Image	234 kB	1	Inf dB	0
Type 15	Input Test Image Vs	Encryption Image	234 kB	0	27.7622326 dB	108.86
		Decryption Image	234 kB	1	Inf dB	0
Type 16	Input Test Image Vs	Encryption Image	234 kB	0	27.7600431 dB	108.91
		Decryption Image	234 kB	1	Inf dB	0


The Size of the Image, isequal() Function, PSNR and MSE are taken as parameters and those things are compared among the input image, encrypted image and the decrypted image in DIOT and SIAT. The above mentioned parameters have been proved that the proposed algorithms are encrypted and decrypted the testing images and also the same parameters are proving that, there are differences presented between the sixteen different encryption algorithm's encrypted images in SIAT. The PSNR Value and MSE Rate equations used for calculations are shown below.

$$\frac{\sum M,N\left[I_2(m,n)-I_2(m,n)\right]^2}{M*N} \quad (1)$$

$$10\log_{10}\left(\frac{R^2}{MSE}\right) \quad (2)$$

The Equation 1 is used to calculate the MSE Rate and the Equation 2 is used to calculate the PSNR Value.

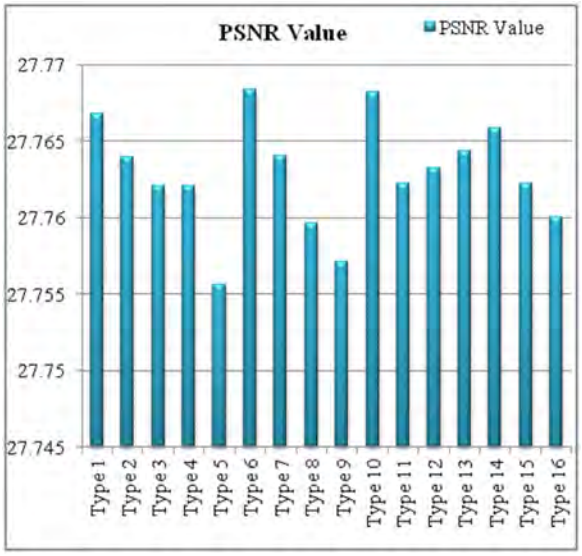


Figure 10. PSNR Differences between 16 Algorithms' Output

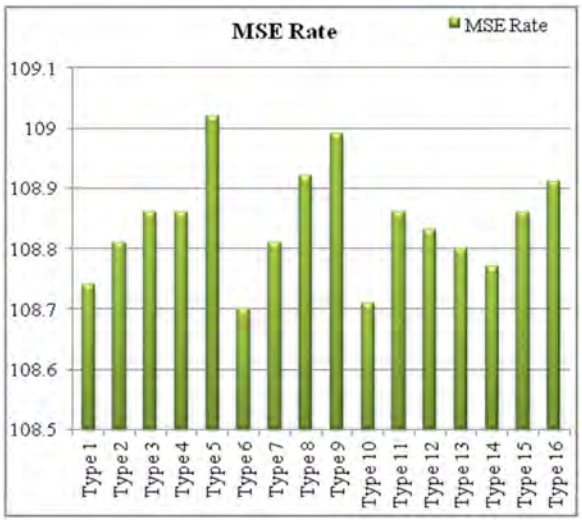


Figure 11. MSE Differences between 16 Algorithms' Output

Figure 10 shows the differences in PSNR value among the sixteen different proposed encryption algorithms. Figure 11 shows the differences in MSE rate among the sixteen different proposed encryption algorithms. The figure 10 and figure 11 details proven that, there are differences presented among the sixteen proposed algorithms and the encrypted images are different from one another algorithm's output.

6. Conclusion

The users' rights and privacy should not to be affected in online image encryption, and the users' image need to be encrypted within the users' country border limit. Moreover, the encrypted images needed to be kept back within users' country border limits and finally the users and their service provider who need to maintain the trust between them are the most important highlighted requirements of the cloud service users. Those things are covered in the Secured Cloud Data Storage Prototype Model and in that proposed model the image's confidentiality will be taken care of by Multi-Dimensional Encryption and Decryption Model. The proposed method has encrypted and decrypted the test images successfully in both multiple image single method testing and single image all propose algorithm methods testing. The single image all proposed algorithm methods testing has proved that all the 16 different encryption methods are not the same; each and every method will provide different encrypted image. The algorithm testing has been done on 512 x 512 pixel images only. In future the proposed algorithm will be extended to handle any size of images for encryption and decryption process to maintain the confidentiality of the image in online.

7. References

[1] https://en.wikipedia.org/wiki/A_picture_is_worth_a_thousand_words, Date of Accessed: 09/08/2017

[2] D. Desai1, A. Prasad, J. Crasto, Chaos-Based System for Image Encryption, International Journal of Computer Science and Information Technologies, Vol. 3 , No. 4, pp. 4809-4811, 2012.

[3] K. Sakthidasan and B. V. Santhosh Krishna, A New Chaotic Algorithm for Image Encryption and Decryption of Digital Color Images, International Journal of Information and Education Technology, Vol. 1, No. 2, pp. 137 – 141, 2011.

[4] N. Agarwal, H. Sharma, An Efficient Pixel-shuffling Based Approach to Simultaneously Perform Image Compression, Encryption and Steganography, IJCSMC, Vol. 2, No. 5, pp. 376 – 385, 2013.

[5] Q-A Kester, Image Encryption based on the RGB PIXEL Transposition and Shuffling, International Journal of Computer Network and Information Security, Vol. 7, pp. 43-50, 2013.

[6] J. Zhao, W. Guo, R. Ye, A Chaos-based Image Encryption Scheme Using Permutation-Substitution Architecture, International Journal of Computer Trends and Technology, Vol. 15 No. 4, pp. 174 – 185, 2014.

[7] D. Lohit Kumar, Dr. A.R.Reddy, Dr.S.A.K.Jilani, Implementation of 128-bit AES algorithm in MATLAB", International Journal of Engineering Trends and Technology, Vol. 33 No. 3, pp. 126 – 129, 2016.
[8] http://www.oxforddictionaries.com/definition/english/cloud-computing, Date of Accessed: 05/05/2017.
[9] D. Boopathy and Dr. M. Sundaresan, Securing Public Data Storage in Cloud Environment, ICT and Critical Infrastructure: 48th Annual Convention of Computer Society of India, Visakhapatnam, India, pp.555-562, 2013.
[10] D. Boopathy and Dr. M. Sundaresan, Secured Cloud Data Storage – Prototype Trust Model for Public Cloud Storage, International Conference on Information and Communication Technology for Sustainable Development, Ahmadabad, India, pp.329–337, 2015.
[11] D. Boopathy and Dr. M. Sundaresan, Framework Model and Algorithm of Request based One Time Passkey (ROTP) Mechanism to Authenticate Cloud Users in Secured Way, 3rd International Conference on Computing for Sustainable Global Development, New Delhi, India, pp.5317–5322, 2016.
[12] D. Boopathy and Dr. M. Sundaresan, A Framework for User Authentication and Authorization using Request based One Time Passkey and User Active Session Identification, International Journal of Computer applications, Vol.172, No.10, pp.18–23, 2017.
[13] D. Boopathy and Dr. M. Sundaresan, Data Type Identification and Extension Validator Framework Model for Public Cloud Storage, Big Data Analytics - Proceedings of the 50th Annual Convention of Computer Society of India, New Delhi, India, pp.533–541, 2014.
[14] D. Boopathy and Dr. M. Sundaresan, Data Encryption Framework Model with Watermark Security for Data Storage in Public Cloud Model, IEEE Eighth International Conference on Computing for Sustainable Global Development, New Delhi, India, pp.1040–1044, 2014.
[15] D. Boopathy and Dr. M. Sundaresan, Enhanced Encryption and Decryption Gateway Model for Cloud Data Security in Cloud Storage, Emerging ICT for Bridging the Future - 49th Annual Convention of Computer Society of India, Hyderabad, India, pp.415–421, 2014.
[16] D. Boopathy and Dr. M. Sundaresan, Policy Based Data Encryption Mechanism Framework Model for Data Storage in Public Cloud Service Deployment Model, Elsevier Fourth International Joint Conference on Advances in Computer Science , Haryana, India, pp.423–429, 2013.
[17] D. Boopathy and Dr. M. Sundaresan, IDOCA and ODOCA – Enhanced Technique for Secured Cloud Data Storage, International Journal of Intelligent Engineering and Systems, Vol.10, No.06, pp. 49 - 59, 2017.
[18] https://en.wikipedia.org/wiki/Pixel, Date of Accessed: 05/11/2017.
[19] https://en.wikipedia.org/wiki/Standard_test_image, Date of Accessed: 05/11/2017.
[20] https://en.wikipedia.org/wiki/TIFF, Date of Accessed: 11/11/2017.
[21] https://en.wikipedia.org/wiki/Image_file_formats, Date of Accessed: 11/11/2017.
[22] http://in.mathworks.com/help/matlab/ref/isequal.html, Date of Accessed: 18/11/2017.
[23] https://en.wikipedia.org/wiki/Peak_signal-to-noise_ratio,Date of Accessed: 18/11/2017.
[24] https://en.wikipedia.org/wiki/Mean_squared_error, Date of Accessed: 18/11/2017.
[25] https://en.wikipedia.org/wiki/Histogram, Date of Accessed: 20/11/2017.

Biographies

D.Boopathy is a Research Scholar doing his PhD in Computer Science in the Department of Information Technology at Bharathiar University. He is qualified with M.Sc.(IT) and MCA from Bharathiar University. He did his Master of Philosophy in Computer Science at Dr. G.R.D College of Science, Coimbatore. His areas of interests are Information Security, Data Privacy and Cloud Computing. He is a Life Member of Computer Society of India and Indian Science Congress Association. So far he has co-authored 2 book chapters for 2 edited books (2 for IGI Global USA).
Email ID: ndboopathy@gmail.com

Prof.Dr.M.Sundaresan is currently Professor and Head of the Department of Information Technology at Bharathiar University, Coimbatore, India. He holds PhD in computer science. He has contributed more than 50 research papers in different areas of Computer Science such as Image Processing, Data Compression, Natural Language Processing, Speech Processing and Cloud Computing in reputed journals. He is a Senior and Life Member of Professional Bodies such as Computer Society of India, Indian Science Congress Association, and Indian Society for Technical Education and IACSIT. He is also in editorial board of five journals. He is the Sectional President for Information and Communication Science & Technology (including Computer Sciences) section in 105th Indian Science Congress. He is the Regional Vice President for Regional VII in Computer Society of India. So far he has authored 2 book chapters for 2 edited books (2 for IGI Global USA). Email ID: bu.sundaresan@gmail.com

www.icgst.com

Rotational invariant Real Time Text Recognition

M. Anyayahan, M. Balinas, A. La Madrid, M. Laurel, C. Lopez, R. Tolentino
Electronics Engineering, Polytechnic University of the Philippines – Santa Rosa Campus, Laguna, Philippines
marron.jann.21@gmail.com balinaskelaine@gmail.com aldrinlamadrid@gmail.com
mark.laurel1595@gmail.com teljoylopez@gmail.com kenmetara@yahoo.com
http://www.pup.edu.ph

Abstract
In everyday life, people always encounter different text images. These text images are in a style of linear or multi-oriented texts in either printed or written form. Due to different orientations of texts in an image, it is a challenge in Optical Character Recognition to recognize this kind of text. In this paper, real time recognition of text in different rotational variations is presented. The performance is done from acquisition of image by a camera and processed by Microsoft Visual Studio. The detection and recognition of text with different rotational variations are achieved by detecting and computing the direction and angle of tilt respectively through the use of geometric and trigonometric principles then recognized by Tesseract optical character recognition engine after counter rotation.[1]

Keywords: *multi orientation angle, rotational variation, tilt angle, tilt direction, Optical Character Recognition.*

Nomenclature

OCR	Optical Character Recognition
BLOB	Binary Large Object
ROI	Region of Interest
CC	Character Confidence
WC	Word Confidence

1. Introduction
Text is a human-readable sequence of characters and the words they form are in either written or printed work. These characters are often in the form of alphanumeric that created series of words. Reading text is a part of our everyday lives. But these texts are not always in a horizontal manner that humans usually see and easily read. Different orientations of text existed due to the creativity of humans, and these text arranged in different orientations can also be certainly read by humans because of their perception. But detection and recognition of these texts in different orientations is a challenge in the field of machine vision.

Numerous studies have been conducted to advance the recognition of text in multi-orientation. The study focuses on end-to-end real-time text localization and recognition method. They present that the real-time performance is achieved by posing the character detection problem as an efficient sequential selection from the set of Extremal Regions. All of the features are scale-invariant, but not all are rotation-invariant however, the features are somewhat robust against small rotations [3]. Another is a proposed technique to extract text from natural scene images but the proposed system is sometimes not able to detect and extract text properly because of some factors like the image may be tilted, some shadow area or the background is complex [1].

A recent study entitled Text-Line Detection, Segmentation and Recognition in Natural Scene presented scene text detection and extraction from images and an algorithm which involves pre-processing of images by applying wiener filter and run length method to detect the text in images. This algorithm does not only detect the text in image but it also detects the blur text. The problem with this study is the certain limitations stated that text with multi-orientation angle cannot be detected [4].

To solve this problem, a system is proposed by the researchers to recognize text with different rotational variations by detecting and computing the direction and angle of tilt respectively through the use of some geometric and trigonometric principles then implementing Optical Character Recognition after counter rotation.

This research is essential to aid the existing studies in advancing image processing. The vital part is to make it more efficient to read text on different rotation variations and to present a new method for detecting tilt direction and angle in text characters. This can also be useful in further studies or development of study about tilt direction and tilt angle in character recognition.

The remainder of the paper is organized as follows: Section (2) focuses on how the system is implemented and evaluated. Section (3) emphasize the results in identifying the direction and angle of tilt and the evaluation result of the system's reliability.

2. Theory
The system is implemented and evaluated using a computer with an Intel core i3 (2GHz) microprocessor with 4 GB RAM running at Windows 10 Home 64-bit Edition, and the sensor used is A1 Tech AW-06 Webcam

with a resolution of 640x480 pixels (30 frames per second). The starting distance is 35 cm. The samples are printed in a 8.5"x11" in Calibri font style and 72 font size. Figure 1 shows the whole process of the system. Text image acquired by the camera is considered as the input image and the one that will be processed by the system.

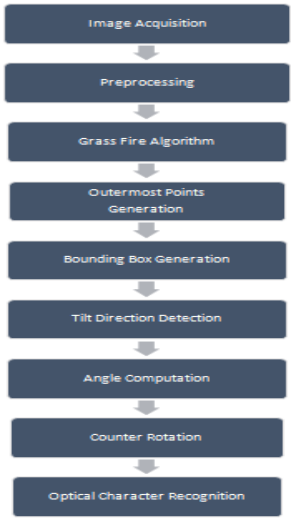


Figure 1 Conceptual Framework

Figure 1 shows the whole process of the system. Text image acquired by the camera is considered as the input image and the one that will be processed by the system. After the image has been acquired, it will be subjected to image pre-processing. Pre-processing includes image binarization and canny edge detection. Image pre-processing is done to make the image free to noise and be converted to full binary image. After pre-processing, Grass Fire Algorithm will be implemented. Grass fire Algorithm works by burning the pixel from a certain point or points to another. In this process, the pixel burning will start from a seed point of a Region of Interest up until the entire region of interest is covered. After pixel burning, all the pixel points that lie on the edge most part of the burned region will be stored in its knowledgebase as outermost points. Then, bounding box will be generated, tracing the mean of values from the outermost points, making the system capable of drawing tilted bounding box. After generating the bounding box, the direction of tilt and its angle will be determined. Counter rotation of the image will be implemented next, considering the direction detected and the angle computed. Lastly, OCR engine will be used to recognize text characters.

A. Identifying the Direction and Angle of Tilt

First step of the whole process of the system is the text image acquisition. This process is done by a camera. Then, the source image will be subjected to pre processing that includes binarization and edge detection to make the image be in pure black and white and to remove noise. After pre-processing, grass fire algorithm will be implemented. In this process, the algorithm starts the pixel burning at a seed point and then, it will spread out to the entire region of interest that covers that seed point that is why it is called region growing because all the pixel that cover the seed point will be selected as part of a new region [4][5]. Figure 2 illustrates pixel burning.

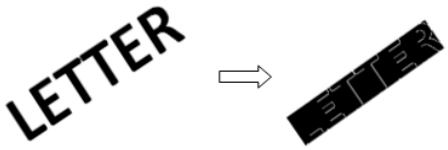


Figure 2 Pixel Burning

The information about the pixel burned are placed on a list or stored in a memory thus making the information about the outermost points be isolated. The system will get the mean of values from the outermost of points firstly generated by the algorithm. Those mean of values will be used to create the straight lines which will lead to generation of the tilting bounding box, which makes it the most essential part of the system. The smallest possible bounding box will enclose the word with all the mean of values of outermost points considered as seen on figure 3. With these tools, the system will be able to create bounding box for the words that are inclined.

Figure 3 Bounding Box

After that, significant points will be derived to be used on detecting and computing the direction and angle of tilt respectively. Since the image is composed of pixels supposedly lying on the Cartesian plane, and the bounding box has been already generated, some information about the bounding box can be established. The bounding box generated is a rectangle consists of two longer sides, two shorter sides and four corner points with x and y coordinates. Significant points will always be the endpoints of the longer side with lower y coordinate as seen on Figure 4.

Figure 4 Significant Points

There is a special tilt case that the system will encounter wherein both of the longer side of bounding box has the same lowest Y coordinate. In this case, no significant points will be established; therefore, no direction detection and angle computation will happen because the system assumes that the image is tilted at 90 degrees making it to be subjected immediately to rotation (90 degrees, clockwise). After establishing the significant points, a decision making process seen on figure 5 will be used on detecting the direction of tilt by comparing its y coordinates. Then, a reference triangle will be drawn to get the angle of tilt through the use of the formula 1 shown on figure 6.

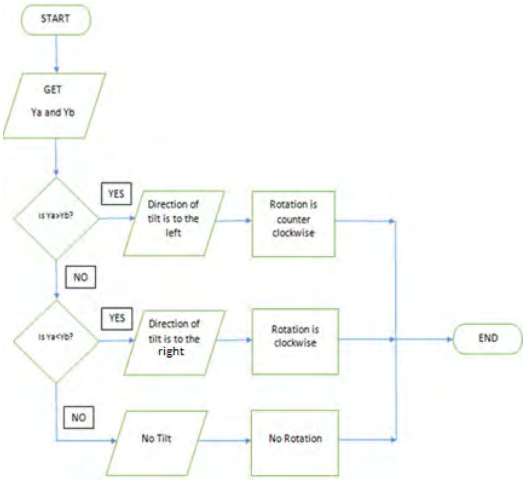


Figure 5 Direction Detection Decision flow

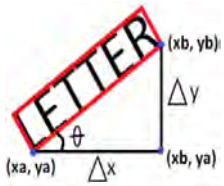


Figure 6 Computation of Angle of Tilt

$$\theta = \tan^{-1}\frac{\Delta y}{\Delta x} \quad \textbf{(1)}$$

After detecting and computing the direction and angle of tilt, counter rotation will be implemented to make the image be back to zero degree orientation. Then, Tesseract Optical Character Recognition will be used together with its confidence function [6]. From the scale of 0-9 with 0 being the best and 9 being the worst, Tesseract OCR engine make judgment on how confident it is that the character recognized is really t he correct character. Then, those values will be fed to formula 2 for the word confidence computation. The system implemented OCR twice thus computing the word confidence also twice, on the word's zero degree orientation and on its 180 degree counterpart as seen on figure 7.

$$Word\ Confidence = \frac{((10-CC1)+(10+CC2)+(10+CC3)+\cdots+(10+CCn))}{10n} \quad (2)$$

CC – Character Confidence

n – Number of Characters in the Word

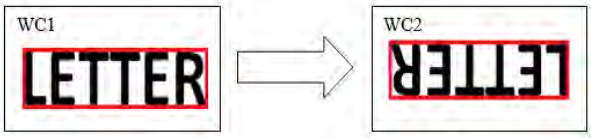


Figure 7 Stages of Word Confidence Reading

After computing the word confidences, it will be used to decide for the output recognition. The output recognition will always be the word with higher confidence.

B. Acquiring the system's reliability on recognition of every rotated text characters in different rotational variations

To determine the reliability of the system, each sample will be subjected in eight different tilt cases to see if there is a significant variation in recognition for each tilt case. Tilt cases are as follows: Case 1 on zero degrees, case 2 on 45 degrees, case 3 on 90 degrees, case 4 on 135 degrees, case 5 on 180 degrees, case 6 on 225 degrees, case 7 on 270 degrees, case 8 on 315 degrees. Correct recognition will tell that the rotation done is right and will be marked as success rotation and recognition. Success rate for each tilt cases will be computed as seen in formula 3.

$$\text{Success Rate(per tilt case)} = \frac{total\ number\ of\ successful\ rotations}{total\ number\ of\ samples} x100 \quad (3)$$

For the overall reliability, the researchers will get the average of all success rates as seen in formula 4.

$$Reliability = \frac{\sum Success\ Rate\ (per\ tilt\ case)}{8} \quad (4)$$

3. Results and Discussion

A. Result of Identifying the Direction and Angle of Tilt

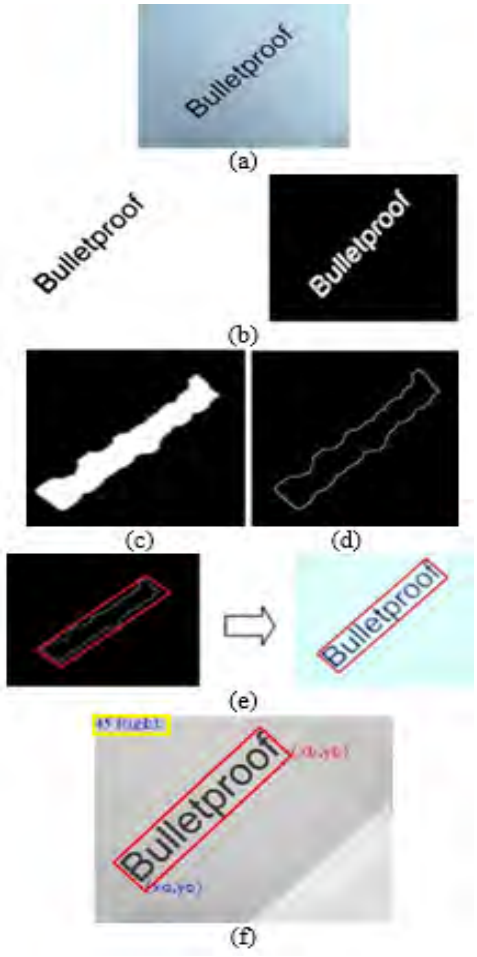


Figure 8 Data Outputs involved in Identifying the Direction and Angle of Tilt (a) Source Text Image (b) Pre-Processed Image (c) Output After Pixel Burning (d) Output After Isolating Outermost Points (e) Bounding Box Generated (f) Direction Detected and Angle Computed

Results are gathered from the 100 samples prepared by the proponents. Figure 8 shows the data outputs of the processes involved in identifying the direction and angle of tilt. Figure 8 (a) shows the source image from one of the samples and (b) shows the output images after pre-processing. Figure 8 (c) shows the output image after pixel burning has been done wherein the white part represents the region of interest burned entirely while Figure 8 (d) shows the output image after the outermost points has been isolated from the region of interest represented by the irregular white line. Figure 8 (e) shows the output images after the bounding box has been generated from the outermost points represented by the red box and Figure 8 (f) shows the output image after the direction and angle of tilt has been identified which, from that specific sample, is 45 degrees to the right written in blue font color. After identifying the direction and angle of tilt, rotation is implemented wherein the image will be rotated as per the angle computed in contrast to the direction identified to make the image be in 0 degree orientation before recognition.

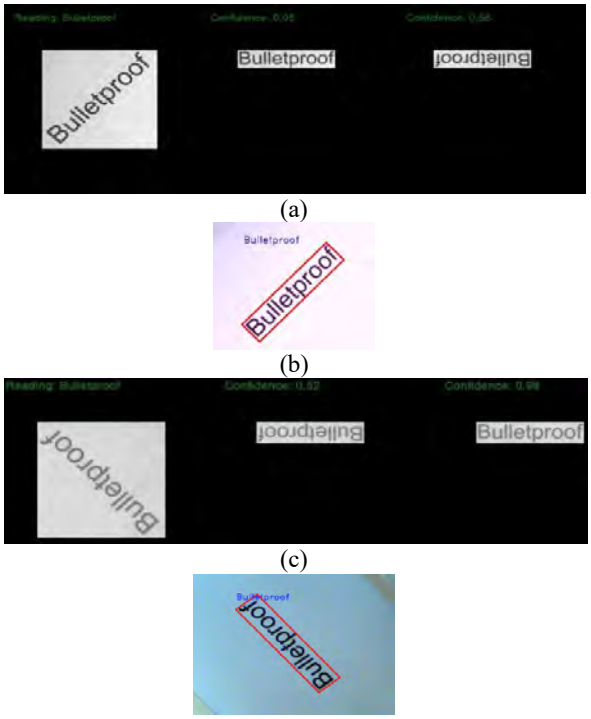


Figure 9 Data outputs After Recognition and Word Confidence Reading (a) WC1 greater that WC2 (b) Output Recognition (c) WC1 less than WC2 (d) Output Recognition

Figure 9 shows the data outputs specific to word confidence reading process as well as the data outputs for recognition after comparing the word confidence readings. Figure 9 (a) shows the stages of word confidence reading when WC1 is greater than WC2. As seen, the reading on first stage is 0.95 while the reading on the second stage is 0.56 making the system decide the output to be on the first stage reading as seen on Figure 9 (b). As seen on figure 9 (c) the reading of word confidence on the second stage is 0.96 which is higher than the word confidence reading on the first which is 0.52, thus, making the output recognition took place on the second stage as seen on figure 9 (d). Figure 10 shows the data outputs of recognition one of the samples subjected to eight (8) different tilt cases.

Figure 10 Data Outputs of Recognition for every Tilt Case (a) Case 1; 0 degree (b) Case 2; 45 degrees (c) Case 3; 90 degrees (d) Case 4; 135 degrees (e) Case 5; 180 degrees (f) Case 6; 225 degrees (g) Case 7; 270 degrees (h) Case 8; 315 degrees

B. Acquired system's reliability on recognition of every rotated text characters in different rotational variations

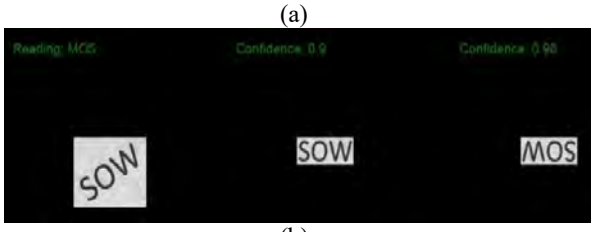
TILT CASES	SUCCESS RATES (%)
1 (0 Degree)	98
2 (45 Degrees)	97
3 (90 Degrees)	95
4 (135 Degrees)	90
5 (180 Degrees)	91
6 (225 Degrees)	90
7 (270 Degrees)	94
8 (315 Degrees)	95
RELIABILITY:	**93.75%**

Table 1 Results of the Study

From 100 samples, success rate is computed in each tilt cases and provided the following outputs: 98% for the first tilt case, 97% for second tilt case, 95% for third tilt case, 90% for the fourth tilt case, 91% for the fifth tilt case, 90% for the sixth tilt case, 94% for the seventh tilt

case, 95% for eighth tilt case. The overall reliability of the system in terms of recognizing every rotated text characters is 93.75%. Table 1 shows the summary of the success rates computed per tilt case and the reliability of the system. As seen, cases 4, 5 and 6 has the lowest success rate due to its first rotated image resulting always to 180 degrees orientation. When optical character recognition is implemented, it calculates the word confidence on the first rotation making the higher chance of getting a higher word confidence than the second word confidence reading. An error rate of 6.25% was also determined. The error rate is consists of word misjudgment and word rearrangement errors. Figure 11 shows the data outputs for misjudgment error. As seen on figure 11 (a) the sample "SOW" was recognized as "MOS". This means that the recognition took place on the 180 degree orientation of the word because the 180 degree orientation of the word formed another word with higher word confidence value than the original word that confuses the system. Figure 11 (b) shows the word confidence on two stages of rotation. As seen on the figure, the sample on the first rotation has an output word confidence of 0.9 and an output on second rotation of 0.96 which is higher than the first reading. Because of that, the output of the system is the second stage of recognition which is the 180 degree counterpart of the sample. This kind of error is usually present on some word, and on some tilt cases depending on the combination of the characters inside the word. The error sometimes happened due to the varying rotation, and sometimes, due to the combination of the characters in the word solely. This error frequently happened on cases 4, 5, and 6.

(a)



(b)

Figure 11 Data Outputs for Misjudgement Error (a) Word Confidence Reading (b) Output Recognition

Figure 12 Data Output for Word Rearranging Error

Figure 12 shows the data output for word rearrangement error. As seen, the sample "die pick die place die attach" were rearranged during recognition when subjected to rotational variations becoming "attach place die pick die die". This happens because the system is reading the recognized word from top to bottom, left to right, disregarding the arrangement of the words when subjected to rotation.

4. Conclusion

The researchers developed a system that can recognize text in different rotational variations by obtaining the right orientation of the text through acquisition of the direction and angle of tilt using geometric and trigonometric principles. Even though the reliability of the system is high, there are still some incidents that the system fails to recognize the text properly. First is when a word, when rotated 180 degree, will result to combination of new characters forming another word, causing the confusion of the system in choosing between the two words from the recognition of two rotated images. This happens frequently on cases 4, 5, and 6 because of the fact that the word being process first is upside down. Second is caused by multiple line of words that when subjected to rotational variations, words were being rearranged, producing an output of disordered words. This happens because the system read the recognizes words from top to bottom. Regardless of the mentioned incidents where errors occurred, the proponents conclude that the system can still detect and recognize text in different rotational variations.

5. Acknowledgements

The proponents wish to express their deepest appreciation and respect to the following who gave possibility to complete this study, without them this could not have been done.

Firstly, the proponents would like to express their deepest faith and gratefulness to the Almighty God whose presence is always been there to guide them and continuously bless them.

Second, to their beloved families and relatives, who support and encourage them unconditionally in spite of their flaws, and are still there to provide for them financial, moral support, lots of love and boundless understanding.

Third, to their Thesis adviser, Engr. Roselito E. Tolentino, for his absolute support, advices, guidance, comments, suggestions, and provisions that benefited them in the completion and success of this study.

Lastly, the proponents would also like to extend their gratitude to their friends and families who have given and shared with them their laughter during times of frustration

References

[1] Kaur, T. and Neeru, N. Text Detection and Recognition from Natural Scene. 2015

[2] Moeslund, T. B. Introduction to Video and Image Processing: Building Real Systems and Applications. 2012

[3] Neumann and Matas. Real-time Lexicon-free Scene Text Localization and Recognition. 2015

[4] Shilpa K.M., Shankar R., Suma Swamy. Text-Line Detection, Segmentation and Recognition In Natural Scene. 2017

[5] Smith, Ray. An Overview of the Tesseract OCR Rngine. 2007

[6] Yun-Kyoo Ryoo, Chan-Myeong Han, Ja-Hyo Ku, Dae-Seong Jeoune,and Young-Woo Yoon. Grassfire Spot Matching Algorithm in 2-DE. International Journal of Bio-Science and Bio-Technology. 2013

Biographies

Marron Jann M. Anyayahan was born in Manila, Philippines on January 5, 1996. He is currently taking up Bachelor of Science in Electronics Engineering at Polytechnic University of the Philippines – Santa Rosa Campus. He was a part of Excelitas Technologies Philippines, Inc. at Cabuyao, Laguna in 2015 and Asian Vision Cable Holdings Inc. at Makati City in 2016 for his On the Job Training. Marron Jann M. Anyayahan is currently a member of Association of Electronics and Communications Engineering Students and was a member of Institute of Electronics Engineers of the Philippines Laguna Chapter.

Ma. Kelaine B. Balinas was born in Carmona Cavite, Philippines on June 13, 1996. She is currently taking up Bachelor of Science in Electronics Engineering at Polytechnic University of the Philippines, Santa Rosa Campus. She was a part of Amkor Technology Philippines at Binan, Laguna in 2015 and Total Power Box Solutions, Inc. at Silang, Cavite in 2016 for her On the Job Training. Ma. Kelaine B. Balinas is currently a member of Association of Electronics and Communications Engineering Students and was a member of Institute of Electronics Engineers of the Philippines Laguna Chapter.

Aldrin P. La Madrid was born in Sta. Rosa Laguna, Philippines on February 18, 1996. He is currently taking up Bachelor of Science in Electronics Engineering at Polytechnic University of the Philippines, Santa Rosa Campus. He was a part of GF Micro Optics Philippines, Inc. at Binan, Laguna in 2015 and PLDT Las Pinas – Alabang Customer Service Operations Zone at Muntinlupa, Metro Manila in 2016 for his On the Job Training. Aldrin P. La Madrid is currently a member of Association of Electronics and Communications Engineering Students and was a member of Institute of Electronics Engineers of the Philippines Laguna Chapter.

Mark Anthony T. Laurel was born in Manila, Philippines on September 15, 1995. He is currently taking up Bachelor of Science in Electronics Engineering at Polytechnic University of the Philippines, Santa Rosa Campus. He was a part of Cirtek Electronics Inc. at Binan, Laguna in 2015 and PLDT Calamba Branch at Calamba, Laguna in 2016 for his On the Job Training. Mark Anthony T. Laurel is currently a member of Association of Electronics and Communications Engineering Students and was a member of Institute of Electronics Engineers of the Philippines Laguna Chapter.

Christelle Joy D. Lopez was born in Sta. Rosa, Laguna, Philippines on June 6, 1996. She is currently taking up Bachelor of Science in Electronics Engineering at Polytechnic University of the Philippines, Santa Rosa Campus. She was a part of Sensor Scientific Phils., Inc. at Calamba, Laguna in 2015 and Total Power Box Solutions, Inc. at Silang, Cavite in 2016 for her On the Job Training. Christelle Joy D. Lopez is currently a member of Association of Electronics and Communications Engineering Students and was a member of Institute of Electronics Engineers of the Philippines Laguna Chapter.

Roselito E. Tolentino is a registered Electronics Engineer and IECEP-Member. He is a graduate of B.S. Electronics and Communication Engineering at Adamson University in 2004 under the scholarship of DOST-SEI. He finished his Master of Science in Electronics Engineering Major in Control System at Mapua Institute of Technology under the scholarship of DOST-ERDT. He currently takes up Doctor of Philosophy in Electronics Engineering at the same Institute. He is currently working as a part time instructor at Polytechnic University of the Philippines Santa Rosa Campus and De La Salle University - Dasmarinas. His research interests are more on Robotics and Machine Vision.